

UNIVERSITETET I OSLO
Fysisk institutt

Treghetsnavigasjon med smarttelefon

Hågen Kyllø

11. juni 2012



Forord

Denne oppgaven er den avsluttende delen av min masterutdanning ved Universitetet i Oslo, Universitetssenteret på Kjeller (UNIK).

Jeg vil rette en takk til min internveileder Oddvar Hallingstad for god og forståelsesfull veiledning og oppfølging.

Kjeller, 2012 06 11

Hågen Kyllø

Sammendrag

Denne masteroppgaven er en fortsettelse av masteroppgaven *Treghetsnavigasjon implementert i Android* som ble avsluttet av Sami Hamra ved Universitetet i Oslo 2011 06 27 [11]. I masteroppgaven til Sami Hamra ble det laget en applikasjon for mobiltelefonen *Google Nexus S* fra Samsung. Applikasjonen leser ut sensordata fra mobiltelefonens innebygde treaksede akselerometer og gyrometer og lagrer dette i en fil. Det er logget sensordata fra flere situasjoner, blant annet mens mobiltelefonen føres i en kvadratisk bevegelse i x-y-planet

På bakgrunn av dette har målet vært å lage et program i Matlab som analyserer de loggede data fra Sami Hamras masteroppgave, beregner mobiltelefonens bevegelse for deretter å kunne bruke det til å beregne areal og volum.

I denne masteroppgaven er det først laget en banegenerator som simulerer en kvadratisk bevegelse av en mobiltelefon i x-y-planet og simulering av navigasjonsligninger. Videre er det utarbeidet pseudokode for Kalman-filter.

Innhold

1	Innledning	11
2	Rom, rammer og koordinatsystemer	13
2.1	Fysisk rom og treghetsrom	13
2.2	Affine rom - rommodell og matematisk rom	13
2.3	Vektorrom	13
2.4	Rammer	14
2.4.1	Treghetsramme $[i]$ ECI	14
2.4.2	Jordramme $[e]$ - ECEF	15
2.4.3	Navigasjonsramme $[n]$ <i>ENU</i>	15
2.4.4	Plattform-/legemeramme $[p]$	16
2.4.5	Forenkling til n -ramme	16
3	Matematisk grunnlag	17
3.1	Vektorer	17
3.1.1	Vektorlengde og vektornorm	17
3.1.2	Indreprodukt (skalarprodukt)	17
3.1.3	Kryssprodukt	18
3.1.4	Skjevsymmetrisk form	18
3.1.5	Ortogonalitet	18
3.2	Orientering og Euler-vinkler	18
3.2.1	Elementære rotasjoner	19
3.2.2	321 Euler-vinkler	19
3.2.3	Derivasjon av RKM	20
4	Strapdown treghetsnavigasjonssystem (S-TNS)	23
4.1	Feil- og støykilder	24
4.1.1	Hvit støy	24
4.1.2	Farget støy	25
4.1.3	Bias (nullpunktsfeil)	25
5	Banegenerator	27
5.1	Bevegelse i ett plan	27
5.2	Funksjoner for bevegelse	28
5.3	Pseudokode for deterministisk banegenerator	30
5.4	Simulering av navigasjonligninger	36
5.4.1	Integrasjonsrutine	36
5.4.2	Pseudokode for simulering av navigasjonligninger	37
6	Kalman-filter (KF)	39
6.1	KF har to faser - prediksjon og måleoppdatering	39
6.2	Linearisert Kalman-filter (LKF)	40
6.2.1	Ligninger for LKF	40

6.2.2	Diskretisering	41
6.2.3	Pseudokode for Kalman-filter	42
7	Konklusjon	45
8	Videre arbeid	47
9	Kilder og referanser	49
	Vedlegg A	51
	Vedlegg B	57

Figurer

2.1	Rammer	14
2.2	Sammenhengen mellom rammer	15
3.1	Elementær rotasjon	19
4.1	Blokkdiagram av strapdown TNS	23
5.1	Banegenerator - omløp	27
5.2	Banegenerator - simulert omløp	32
5.3	Banegenerator - simulert for x-akse	33
5.4	akselerasjon simulert med støy	34
5.5	Utsnitt av akselerasjon med støy	35
5.6	Vinkelhastighet uten og med støy	35
5.7	Eulers forovermetode	36
5.8	Navigasjonsligninger simuleringseksempel 1	38
5.9	Navigasjonsligninger simuleringseksempel 2	38
5.10	Navigasjonsligninger simuleringseksempel 3	38
6.1	Blokkskjema av linearisert Kalman-filter	40

Nomenklatur - symboler og forkortelser

i, e, n, p	- forkortelse for treghets-, jord-, navigasjons- og plattformramma
\vec{v}	- geometrisk vektor
\underline{v}	- algebraisk vektor
$S(\omega)$	- skjevsymmetrisk form av ω
$R(\theta)_y$	- rotasjon om akse y med vinkel theta
R_p^n	- rotasjonsmatrisa fra p -ramma til n -ramma
\underline{p}	- posisjonsvektor
\underline{v}	- hastighetsvektor
\underline{a}	- akselerasjonsvektor
\underline{f}	- vektor for spesifikk kraft
θ	- theta: vinkelposisjon
ω	- omega: vinkelhastighet
ω_p^{np}	- vinkelhastigheten i p -ramma sett fra n -ramma representert i p
$p'(t)$	- uttrykk for posisjon i benagenerator
$v'(t)$	- uttrykk for hastighet i benagenerator
$a'(t)$	- uttrykk for akselerasjon i benagenerator
t_0, t_1, \dots, t_8	- tidspunkter i banegeneratoren
VRW	- velocity random walk (hvit støy for akselerometer)
ARW	- angular random walk (hvit støy for gyrometer)
β^a	- bias for akselerometer
β^g	- bias for gyrometer
\underline{x}	- tilstandsvektor
$\hat{\underline{x}}$	- estimert tilstand (a posteriori)
$\bar{\underline{x}}$	- prediktert tilstand (a priori)
\underline{z}	- målevektor
\underline{u}	- pådrag
\underline{v}_k	- prosessstøy
\underline{w}_k	- målestøy
\bar{P}_k	- kovarians til prediktert tilstand
\hat{P}_k	- kovarians til estimert tilstand
K_k	- Kalman-filter-forsterkning
H_k	- målematrise
F	- systemmatrise (kontinuerlig)
L	- Pådragsmatrise (kontinuerlig)
G	- prosessstøymatrise (kontinuerlig)
Φ	- systemmatrise (diskret)
Λ	- pådragsmatrise (diskret)
Γ	- prosessstøymatrise (diskret)

1 Innledning

Rapporten er delt inn i nummererte kapitler. De første kapitlene er en gjennomgang av bakgrunnsstoff som ligger til grunn for denne masteroppgaven. Kapittel 2 tar for seg rom og rammer. Her blir det forklart forskjellen mellom det fysiske rommet hvor en navigasjon foregår, rommet hvor dette modelleres og rammer hvor matematikk blir anvendt. Kapittel 3 går gjennom det matematiske grunnlaget.

I kapittel 4 forklares treghetsnavigasjonssystemet blokkskjematisk samt kort om feil- og støykilder. I kapittel 5 utledes banegeneratoren i detalj. Pseudokoden og figurer fra simulering er vist.

I Kapittel 6 forklares Kalman-filteret generelt. I denne masteroppgaven blir det brukt et linearisert Kalman-filter som forklares i dette kapitlet. Diskretisering og pseudokode blir vist.

Figurer er nummerert etter kapitlene med et etterfølgende tall som sier hvilken figur det er i rekkefølgen i inneværende kapittel. Eksempelvis er *Figur 3.2* er den andre figuren i kapittel 3.

2 Rom, rammer og koordinatsystemer

(Kilder brukt i dette kapitlet er [2], [3], [6] og [7].)

2.1 Fysisk rom og treghetsrom

Et referanserom (eller referansesystem) er et fysisk rom. I navigasjonen er det her fartøyet fysisk befinner seg. Et treghetsrom er et spesielt referanserom hvor Newtons lover gjelder i sin enkleste form. Geometrien er Euklidsk. Relativistiske aspekter i forhold til objekter som beveger seg i forhold til hverandre ses bort fra. I et treghetsrom er tiden homogen, det vil si at klokker i rommet kan synkroniseres. Rommet er homogent og tidsuavhengig, dvs at fartøyet posisjon i rommet ikke påvirker navigasjonen. Alle posisjoner i rommet, retninger og tiden er ekvivalent med hensyn på formulering av bevegelseslovene.

2.2 Affine rom - rommodell og matematisk rom

Affine rom brukes som modell av referanserommet. Det er her objektene blir modellert inn. I det affine rom legges det ut rammer i de forskjellige objektene. I forbindelse med at rammer plasseres ut i de forskjellige objektene spesifiseres origo til de rammene og tilhørende basisvektorer.

I denne oppgaven benyttes en navigasjonsramme og en ramme for mobiltelefonen med origo i mobiltelefonens sentrum. Rammer er koordinatsystemer med origo fiksert i objektenes sentrum. For å få en matematisk representasjon brukes den matematiske enheten vektorrom. Et vektorrom er et matematisk rom hvor objekter er matematisk representert med koordinatsystemer og punkter. Det er i vektorrommet vi anvender vår matematikk. Vi kan matematisk beskrive posisjoner til punkter og deres bevegelser i forhold til hverandre.

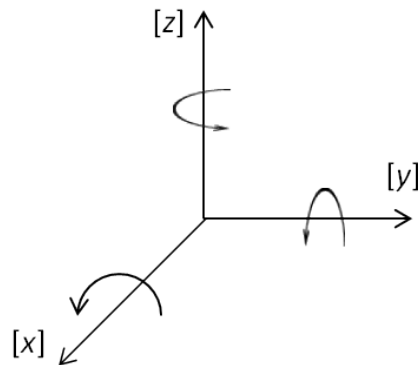
2.3 Vektorrom

Et vektorrom er matematisk konstruksjon. Det er i dette rommet matematikken blir anvendt. En kan på den måten si at vi her opererer vår matematikk for objektene i det affine rommet. Rommet er formet av vektorer som er definert med operasjonene vektoraddisjon og skalarmultiplikasjon. Et eksempel på et vektorrom er et rom av Euklidske vektorer. Euklidske rom kan eksempelvis brukes til å representere fysiske størrelser som krefter. To vektorer som representerer krefter av samme type kan adderes for å gi en resulterende kraft representert av en tredje vektor.

Vektorrom har dimensjoner. For det Euklidske rommet \mathbb{R}^3 brukt i navigasjonssammenheng kan en romlig sett betrakte dimensjonene i vektorrommet som uavhengige retninger i rommet. Det Euklidske vektorrommet \mathbb{R}^2 kan oppfattes som vektorer i et plan.

2.4 Rammer

For rammene i denne oppgaven vil en dreining om aksene i positiv retning tilsvare en dreining med klokka sett fra negativ ende av aksene - høyrehåndsreglen.

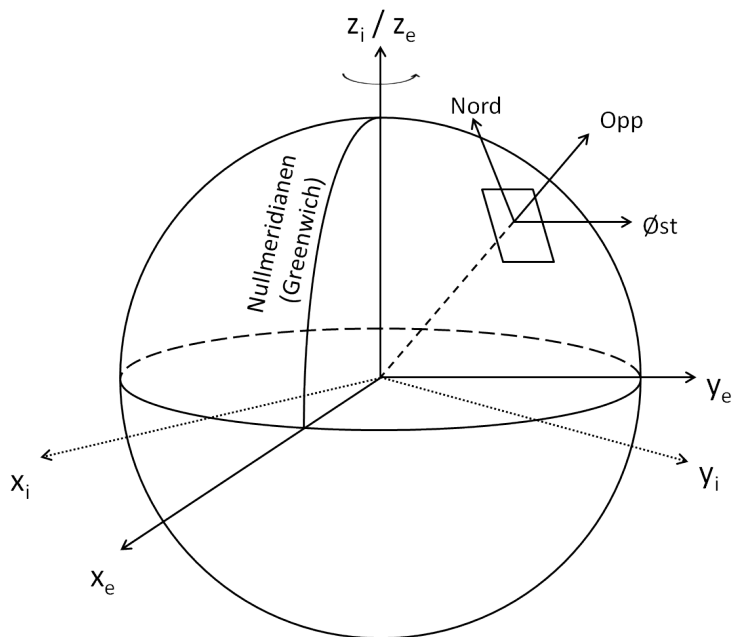


Figur 2.1: Rammer

2.4.1 Treghetsramme [*i*] ECI

i brukes som suffiks for treghetsrammen. Aksenotasjon er x_i , y_i og z_i . Koordinatsystemet til treghetsrammen kalles ECI som står for Earth-Centered Inertial. ECI har origo i jordas sentrum. Treghetsrammen ECI er inertiell i motsetning til ECEF som roterer i treghetsrommet slik at koordinatsystemet står fast i forhold til jordoverflata. ECI står i ro i forhold til fiksstjerner. x-y-planet sammenfaller med ekvatorialplanet til jorda. z-aksen har positiv retning gjennom nordpolen.

ECI er ikke inertiell i egentlig forstand, da jorda akselererer på grunn av sin bane rundt sola og at solsystemet akselererer i galaksen. For de fleste tilfeller vil det allikevel være tilstrekkelig å betrakte ECI som inertiell.



Figur 2.2: Sammenhengen mellom rammer

2.4.2 Jordramme $[e]$ - ECEF

e brukes som suffiks for jordrammen. Aksenotasjon er x_e , y_e og z_e . Koordinatsystemet ECEF står for Earth Centered Earth Fixed. Denne rammen roterer med jorda og har origo i jordas sentrum slik at koordinatsystemet står i ro i forhold til jordoverflata. z -aksen går gjennom polene med nordpolen som positiv retning, x -aksen går gjennom nullmeridianen (Greenwich) og y -aksen kan defineres av høyrehåndsreglen til å gå gjennom ekvator ved 90° lengdegrad. På grunn av jordrotasjonen roterer jordramma i treghetsrommet om treghetsrammens akse x_i .

2.4.3 Navigasjonsramme $[n]$ *ENU*

n brukes som suffiks for jordrammen. Aksenotasjon er x_n , y_n og z_n . Navigasjonsrammen er en lokal geografisk navigasjonsramme som er plassert der hvor navigasjonssystemet befinner seg. Denne rammen ligger tangentielt på jordoverflata og har sine akser plassert på linje mot øst, nord og lokalt vertikalt opp. Ramma følger jordoverflaten og har alltid sin z -akse negativt pekende mot e -rammens origo (jordens sentrum).

2.4.4 Plattform-/legemeramme [p]

p brukes som suffiks for denne rammen. Aksenotasjon er x_p , y_p og z_p . Plattformrammen er fast i fartøyet (mobiltelefonen i denne oppgaven) og dens akser sammenfaller med sensoraksene. Plattformrammen har sine akser definert tilsvarende som for e -rammen.

2.4.5 Forenkling til n -ramme

Sensorene brukt i mobiltelefonen er ikke nøyaktige nok til å måle jordrotasjonen. Støyen fra sensorene vil også overgå endringene i jordrotasjonen. Målingene foregår over korte avstander og få sekunder. For denne oppgaven forenkles derfor jorda til å ha en fiksert posisjon og stilling i rommet og jordoverflaten anses som flat. Dette gjør at i -rammen ses som fiksert i e -rammen og gjør dem sammenfallende. Disse sammenfallende rammene reduseres derfor til en enkelt ramme som blir navigasjonsrammen n .

3 Matematisk grunnlag

(Kilder brukt i dette kapitlet er [1], [2], [3], [6] og [7].)

3.1 Vektorer

I fysikken er en skalar en fysisk størrelse som er endelig beskrevet med et måltall og en måleenhet. Eksempler på skalarer er tid og masse. I matematikken er en skalar et reelt tall.

Noen elementer er ikke endelig beskrevet før de også har en definert retning. En skalar som også har en definert retning kalles en vektor. En vektor beskriver lengde og retning, hvor lengden representerer skalarkomponenten. I fysikken er hastighet og akselerasjon eksempler på vektorer.

Vektorregning inngår som en del av lineær algebra hvor en vektor er et element i vektorrommet. En geometrisk vektor i et 3-dimensjonalt koordinatsystem med basisvektorene b_i kan uttrykkes som

$$\vec{v} = \sum_{i=1}^3 v_i \vec{b}_i$$

hvor vektoren \vec{v} algebraisk kan uttrykkes som en matrise:

$$\underline{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

3.1.1 Vektorlengde og vektornorm

Lengden av en vektor kan skrives $|v|$. For vektorrommet og er definert ved

$$|v| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

En generalisering av en vektorlengde til et vektorrom kalles Euklidsk norm og er definert ved

$$||v|| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

3.1.2 Indreprodukt (skalarprodukt)

Et indreprodukt avbilder to vektorer inn på en skalar og er et mål på forholdet mellom de to vektorene. Et skalarprodukt gir et reelt tall.

$$\langle \vec{v}, \vec{u} \rangle = |\vec{v}| |\vec{u}| \cos \angle \vec{v} \vec{u}$$

Indreprodukt kan brukes til å bestemme om to vektorer er ortogonale, står vinkelrett på hverandre. I det tilfellet vil indreproduktet være lik 0.

3.1.3 Kryssprodukt

Kryssprodukt er en måte å multiplisere sammen to tredimensjonale vektorer på som resulterer i en nye vektor som står vinkelrett på de opprinnelige vektorene. Kryssprodukt er definert ved:

$$|\vec{v} \times \vec{u}| = |\vec{n}||\vec{v}||\vec{u}|\sin\angle\vec{v}\vec{u}$$

\vec{n} er enhetsvektoren og står vinkelrett på begge de opprinnelige vektorene med positiv retning i henhold til høyreregelen. Lengden til den nye vektoren tilsvarer arealet av parallelogrammet de to opprinnelige vektorene spenner ut.

3.1.4 Skjevsymmetrisk form

En skjevsymmetrisk matrise er definert slik at $a_{ij} = -a_{ji}$ og med alle elementene langs hoveddiagonalen lik 0. Den skjevsymmetriske matrisen er kvadratisk og negativt symmetrisk (skjevsymmetrisk) om diagonalen. Den skjevsymmetriske form finnes ved å ta kryssproduktet av

$$s_{ij}^p = \langle (\omega_1 \vec{P}_1 + \omega_2 \vec{P}_2 + \omega_3 \vec{P}_3) \times \vec{P}_j, \vec{P}_i \rangle$$

hvor P er den basisvektoren som danner det ortogonale koordinatsystemet. Den skjevsymmetriske formen til en vektor blir da

$$S(\underline{\omega}^P) = \underline{\omega}^P \times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \text{ hvor } \vec{\omega} = \omega_1 \vec{P}_1 + \omega_2 \vec{P}_2 + \omega_3 \vec{P}_3$$

3.1.5 Ortogonalitet

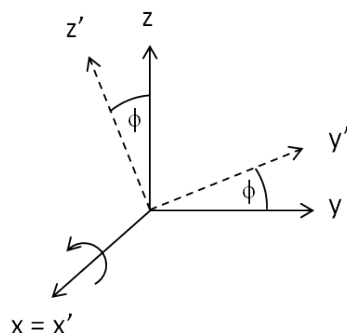
I denne oppgaven brukes ortogonale rammer. Det vil si at basisvektorene for rammene peker i uavhengige retninger i rommet. For to vektorer som står vinkelrett på hverandre kan ikke den ene brukes til å beskrive den andre. De er derfor ortogonale. Om to vektorer er ortogonale kan det finnes med indreproduktet av dem som da må være lik 0.

3.2 Orientering og Euler-vinkler

Orienteringen beskriver vinkelforskjellen mellom to forskjellige rammer. For et fartøy vil fartøyets orientering beskrives som plattformen $[p]$'s vinkling i rommet i forhold til navigasjonsplanet, $[n]$. For tre-dimensjonale navigasjonssystemer er en vanlig måte å oppgi plattformens orientering på er ved bruk av Euler-vinkler.

3.2.1 Elementære rotasjoner

I denne oppgaven brukes ortogonale koordinatsystemer. Euler-vinkler angir vinkelforskjellen mellom to koordinatsystemer. En rotasjon av et koordinatsystem kan beskrives ved tre vinkler. Vinklene gis om basisaksene x , y og z . I navigasjonssammenheng kalles vinkelrotasjonene ofte for roll (rull), pitch (stamp) og yaw (gir), og blir ofte angitt som henholdsvis ϕ , θ og ψ . Rotasjon av en vektor om enkeltakser kan representeres i matriser og kalles elementære retningskosinmatriser (RKM).



Figur 3.1: Elementær rotasjon

Figur 3.1 viser et eksempel på en elementær rotasjonen om x-aksen. De elementære RKM som beskriver rotasjonene om x , y og z er:

$$R(\phi)_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$

$$R(\theta)_y = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}$$

$$R(\psi)_z = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.2.2 321 Euler-vinkler

De elementære rotasjonene kan reduseres til en enkelt retningskosinmatrise (RKM). Det kan gjøres ved å multiplisere sammen de elementære RKM til en resulterende RKM. Med de tre aksene gir det tilsammen 6 forskjellige rekkefølger de kan multipliseres sammen på. I navigasjonssammenheng er det vanlig å bruke 321 Euler-vinkler. 321 kommer av at de elementære RKM

multipliseres sammen i rekkefølgen ψ , θ og ϕ . Det gir den resulterende matrisen:

$${}^E R_b^n = \begin{bmatrix} c_{\theta_3} c_{\theta_2} & c_{\theta_3} s_{\theta_2} s_{\theta_1} - s_{\theta_3} c_{\theta_1} & c_{\theta_3} s_{\theta_2} c_{\theta_1} + s_{\theta_3} s_{\theta_1} \\ s_{\theta_3} c_{\theta_2} & s_{\theta_3} s_{\theta_2} s_{\theta_1} + c_{\theta_3} c_{\theta_1} & s_{\theta_3} s_{\theta_2} c_{\theta_1} - c_{\theta_3} s_{\theta_1} \\ -s_{\theta_2} & c_{\theta_2} s_{\theta_1} & c_{\theta_2} c_{\theta_1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Det er to hovedtyper av rotasjoner; om faste akser eller om nye akser. RKM definert om faste akser skrives ${}^F R_b^n$ og RKM definert om nye akser skrives ${}^E R_b^n$.

Ved faste akser roteres det først en vinkel om en av aksene, deretter roteres det en vinkel om neste akse med utgangspunkt i slik stillingen til ramma var før rotasjonene startet.

Ved nye akser roteres det først en vinkel om en akser, deretter roteres det en vinkel om neste akse med utgangspunkt i slik ramma står etter første rotasjon. Tre rotasjoner om nye akser gir samme sluttstilling som tre rotasjoner om faste akser i motsatt rekkefølge.

3.2.3 Derivasjon av RKM

For to ortonormale rammer \mathcal{F}_p og \mathcal{F}_n er differensialligninga for retningskosinmatrisa R_p^n :

$$\dot{R}_p^n(t) = S(\underline{\omega}_p^n(t))R_p^n(t) = R_p^n(\underline{\omega}_p^{np}(t)) \text{ med } R_p^n(0) \text{ gitt.}$$

Viss R er en ortogonal matrise (R_p^n) har vi:

$$RR^T = I$$

Derivasjon gir:

$$\dot{R}R^T + R\dot{R}^T = \dot{R}R^T + (\dot{R}R^T)^T = 0$$

Viss S defineres som $S = \dot{R}R^T$ så er S en skjevsymmetrisk matrise siden S innsatt i uttrykket over gir $S + S^T = 0$

Dette viser at det er en sammenheng mellom den deriverte av en ortonormal matrise R og matrisa S som har skjevsymmetrisk form:

$$S = \dot{R}R^T = \dot{R}R^{-1} \Leftrightarrow \dot{R} = SR$$

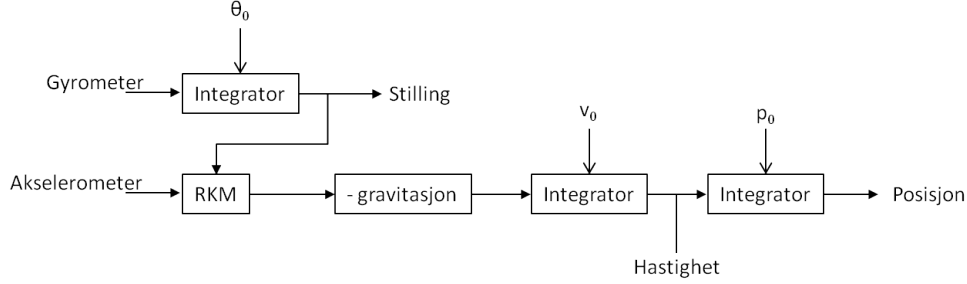
Løsninga av differensialligninga $\dot{R}_p^n(t) = R_p^n(\underline{\omega}_p^{np}(t))$, gitt ovenfor, kan løses dersom en har målinger for $\underline{\omega}_p^{np}$ og $R_p^n(0)$ er kjent. Dette kalles løsningen på det kinematiske problem - sammenhengen mellom vinkelhastigheten og

de deriverte av Euler-vinklene. For dette tilfellet for 321 Euler-vinkler.

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \sin \theta_1 \tan \theta_2 & \cos \theta_1 \tan \theta_2 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \frac{\sin \theta_1}{\cos \theta_2} & \frac{\cos \theta_1}{\cos \theta_2} \end{bmatrix}}_{\text{rotasjonsmatrisa}} \underline{\omega}_p^{np}$$

4 Strapdown treghetsnavigasjonssystem (S-TNS)

(Kilder brukt i dette kapitlet er [2], [6] og [7].)



Figur 4.1: Blokkdiagram av strapdown TNS

I et S-TNS er akselerometrene og gyrometrene montert skrofast i fartøyet. De er montert parallelt med fartøyets legemeakser og i massesentret.

Akselerometrene og gyrometrene i mobiltelefonen er tre-aksede og av typen mikroelektromekanisk system (MEMS). Sensorene er utført som et halvlederdesign med både elektriske og mekaniske egenskaper.

Gyroskopet har elementer som gir utslag når vinkelstillingen endrer seg og gis som vinkelhastigheten om de tre aksene:

$$\vec{g}_x, \vec{g}_y, \vec{g}_z$$

Sensorsignalet fra akselerometret tilsvare akselerasjonen langs akselerometrets tre akser. Viss akselerometret er plassert i massesentret og med akser på linje med fartøyets legemeakser vil akselerometersignalet tilsvare akselerasjonen til fartøyet. Med dobbel integrasjon av signalet vil man kunne finne posisjonen. I et treghetsnavigasjonssystem (TNS) er det ønskelig å referere tilstander som posisjon i henhold til en navigasjonsramme. Med retningskosinmatrisa (RKM) kan akselerasjonsvektoren transformeres til å være representert i n -ramma.

sensorsignalet fra gyrometret gir vinkelhastigheten til fartøyet. Ved en integrasjon av signalet finnes orienteringen. Orienteringen brukes for å finne retningskosinmatrisa.

Av dette gir det navigasjonsligningene:

$$\begin{aligned} \dot{\underline{p}}^n &= \underline{v}^n \\ \underline{v}^n &= R_p^n \tilde{f}^p - \underline{g}^n \\ \dot{R}_p^n &= R_p^n S(\underline{\omega}_p^{np}) \end{aligned}$$

To integrasjoner fører til større problemer med akkumulerte feil.

4.1 Feil- og støykilder

Signal fra akselerometret og gyrometret vil ha støy. Det er ukjent hvilke sensorer som er brukt i mobiltelefonen og verdier brukt for å modellere støy er hentet fra tidligere masteroppgaver som modellerer akselerometre og gyrometre.

4.1.1 Hvit støy

Sensorene antas å ha hvit støy. Det er et stokastisk og ukorrelert signal. Det vil si ta det ikke er mulig å si noe om signalet etter et visst tidspunkt basert på informasjon om signalet før tidspunktet. Hvit støy har 0 i gjennomsnittsverdi og alle frekvenser er likt representert. Videre anses støyen som gaussisk (normalfordelt). For et fysisk system vil båndbredden i praksis allikevel være begrenset. Hvit støy skyldes egenstøyen og kommer av termiske effekter.

For akselerometret kalles hvit støy VRW som er en forkortelse for velocity random walk og kan i datablader oppgis som $\frac{m/s^2}{\sqrt{s}}$ og angir gjennomsnittlig feil. For gyrometret kalles den hvite støyen ARW som er en forkortelse for angular random walk og kan i datablader kan ARW være oppgitt som en konstant med benevnningen $\frac{deg}{\sqrt{h}}$ og angir gjennomsnittlig feil. Random walk-parametrene sier noe om hvordan variansen til systemet øker på grunn av tilfeldig støy fra gyro- og akselerometer.

Feilbidraget fra hvit støy er uavhengig av andre feil. Målingene av fra begge sensorene blir integrert opp. Vinkelhastigheten, $[\omega]$, fra gyroskopet blir integrert opp én gang for å finne stillingen, $[\theta]$. Målingen fra akselerometret, $[f]$, integreres to ganger for å finne posisjonen, $[p]$. Ved integrasjonen blir også den hvite støyen integrert opp og på den måten forsterket. Det at signalet fra akselerometret integreres to ganger gjør at den økte feilen grunnet integrasjon vil være størst for posisjon. Det fører til at systemet vil bli påvirket av den tilfeldige og ukorrelerte hvite støyen. Random walk er feilen grunnet integrasjon av hvit støy. Denne feilen øker med tiden og fører til en stadig økende usikkerhet.

For det diskrete tilfellet er hvit støy definert som en sekvens av stokastiske og ukorrelerte verdier med middelvei lik 0. I oppgaven vil det bli brukt diskrete verdier for hvit støy. I MatLab-programmet gjøres dette ved at det benyttes en funksjon som trekker tilfeldige verdier.

4.1.2 Farget støy

Farget støy inneholder mer energi ved visse deler av frekvensspektret enn andre. I motsetning til hvit støy der alle frekvenser er likt representert, har farget støy en annerledes fordeling og er noe som ikke varierer helt tilfeldig. I denne oppgaven antas det ikke farget støy.

4.1.3 Bias (nullpunktsfeil)

Bias er en konstant feil på signalet. For en sensor som ikke har pådrag vil det komme til uttrykk som et signalutslag forskjellig fra null.

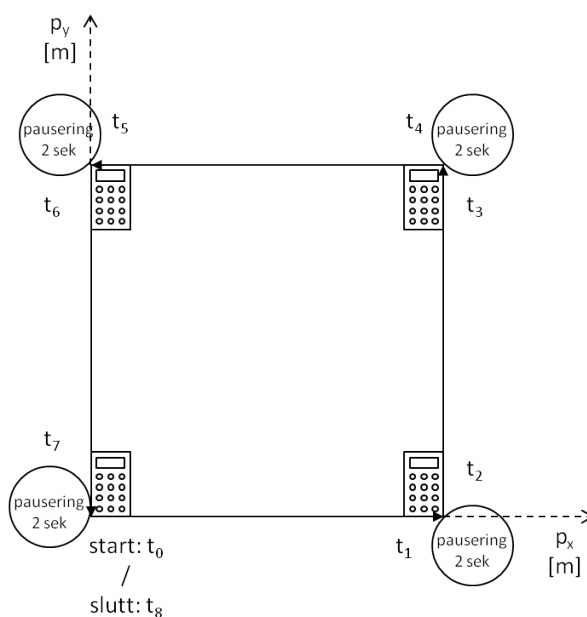
5 Banegenerator

(Kilder brukt i dette kapitlet er [1], [2] og [3])

I Matlab er det laget en banegenerator. Banegeneratoren genererer deterministiske data for bevegelse av mobiltelefonen langs en bane i n -ramma. Det genereres data for spesifikk kraft $[f]$, hastighet $[v]$, posisjon $[p]$ og vinkelhastighet $[\omega]$. Bevegelsen tilsvarer at mobiltelefonen flyttes fra hjørne til hjørne på toppen av en eske som står på et flatt underlag.

5.1 Bevegelse i ett plan

Det simuleres ett omløp av en kvadratisk bevegelse av mobiltelefonen i planet i n -ramma, bestemt av p_x og p_y . Banen består av rettlinjede bevegelser. I hvert hjørne simuleres det at mobiltelefonen står i ro i 2 sekunder. Figur 5.1 viser forløpet.



Figur 5.1: Banegenerator - omløp

Bevegelsen starter ved tiden $t_0 = 0,00$ s og ender etter pausering i siste hjørne ved tiden $t_8 = 16,00$ s. Omløpet på 16 s er stykket opp i 9 deler fra t_0 til t_8 . Banegeneratoren bruker tidsinkrement på 0,01 s.

Forløpet av mobiltelefonens simulerte omløp:

$t_0 - t_1$	0,3 m i p_x -retningen i løpet av to sekunder,
$t_1 - t_2$	er i ro i to sekunder,
$t_2 - t_3$	0,3 m i p_y -retningen i løpet av to sekunder,
$t_3 - t_4$	er i ro i to sekunder,
$t_4 - t_5$	0,3 m i negativ p_x -retning i løpet av to sekunder,
$t_5 - t_6$	er i ro i to sekunder,
$t_6 - t_7$	0,3 m i negativ p_y -retning i løpet av to sekunder og til slutt
$t_7 - t_8$	er i ro i 2 sekunder.

5.2 Funksjoner for bevegelse

Disse ligningene beskriver bevegelsen av mobiltelefonen i n -ramma. Ved tidspunkt t_0 sammenfaller mobiltelefonens plattformramme med navigasjonsramma. Ømløpet starter i origo. For hastighet har den simulerte bevegelsen langs linjestykkene et klokkeformet forløp med topphastigheten halvveis i forløpet. Hastigheten er 0 i start- og sluttidspunktene. Det beregnes simulerte måleverdier med intervaller på 0,01 sekunder.

Generelt er:

$$\begin{aligned}\underline{p}^n(t) &= \iint \underline{f}^n \\ \underline{v}^n(t) &= \int \underline{f}^n \\ \underline{a}^n(t) &= \underline{f}^n \\ \underline{\theta}_b^n &= \int \underline{\omega}^n\end{aligned}$$

Spesifikk kraft $[f]$ er definert som ikke-gravitasjonell kraft per enhetsmasse og måles i $\frac{m}{s^2}$:

$$\text{spesifikk kraft} = f = \frac{F_{(\text{ikke-gravitasjonell})}}{\text{masse}}$$

Newtons 2. lov er $F = ma$ som gir at $a = \frac{F}{m}$. Dette gir spesifikk kraft som $f = a$. Har da et uttrykk for spesifikk kraft som kan brukes for å finne et uttrykk for akselerasjonen langs banen.

Uttrykk for akselerasjon

$$\begin{aligned}\underline{a}'(t) &= f \sin(\omega t)|_{t_0}^{t_1} \\ &= f(\sin(\omega t_1) - \sin(\omega t_0))\end{aligned}$$

Uttrykk for hastighet

Hastigheten ved starttidspunktet er 0 slik at v_0 blir lik 0.

$$\begin{aligned}
\dot{\mathbf{v}}'(t) &= a'(t) \\
v'(t) &= v_0 + \int_0^t f \sin(\omega t) dt \\
&= 0 + \left(-\frac{f}{\omega} \cos(\omega t) \right) \Big|_{t_0}^{t_1} \\
&= -\frac{f}{\omega} [\cos(\omega t_1) - \cos(\omega t_0)]
\end{aligned}$$

Uttrykk for posisjonsendring

Posisjonen ved starttidspunktet er 0 slik at p_0 blir lik 0.

$$\begin{aligned}
\dot{\mathbf{p}}'(t) &= v'(t) \\
p'(t) &= p_0 + v_0 + \int_0^t \frac{f}{\omega} (1 - \cos(\omega t)) dt \\
&= 0 + 0 + \frac{f}{\omega} \left(t - \frac{1}{\omega} \sin(\omega t) \right) \Big|_{t_0}^{t_1} \\
&= \frac{f}{\omega} \left[\left(t_1 - \frac{1}{\omega} \sin(\omega t_1) \right) - \left(t_0 - \frac{1}{\omega} \sin(\omega t_0) \right) \right]
\end{aligned}$$

I og med at omløpet er kvadratisk er bevegelsen lik langs alle fire sider, med unntak av fortegn og langs hvilken akse bevegelsen foregår. Ligningene over kan derfor brukes til å beskrive bevegelsen langs alle de fire sidene i omløpet.

For å få den ønskede bevegelsen er det lagt en sinus på akselerasjonen. Når uttrykket integreres opp gir det et klokkeformet hastighetsforløp med et s-formet posisjonsforløp. I løpet av bevegelsen langs en side skal det ha gått en full syklus. Uttrykket i parentes etter sinus i uttrykket skal derfor være lik 2ω ved tiden t_1 .

Beregner ω for bevegelsesuttrykkene

$$\omega 2 = 2\pi \rightarrow \omega = \pi$$

Beregner f

$$p(t_1) = 30 \text{ cm}$$

$$\begin{aligned}
\frac{f}{\omega} \left(t - \frac{1}{\omega} \sin(\omega t) \right) &= 30 \\
\frac{f}{\omega} &= \frac{30}{\left(t - \frac{1}{\omega} \sin(\omega t) \right)}
\end{aligned}$$

$$f = \left(\frac{30}{2 - \frac{1}{\pi} \sin(\omega 2)} \right) \omega = 47,1238898... \frac{\text{cm}}{\text{s}^2}$$

Uttrykk for vinkelhastighet

Vinkelhastigheten settes til 0 gjennom hele omløpet.

5.3 Pseudokode for deterministisk banegenerator

$t_0 = 1$; $t_1 = 200$; $t_2 = 400$; $t_3 = 600$; $t_4 = 800$; $t_5 = 1000$; $t_6 = 1200$;
 $t_7 = 1400$; $t_8 = 1600$;

```

Ts = 0.01;           % Tidssteg
N = t8;              % Antall tidssteg
t_vec = (0 : N) * Ts % Tidsvektor
f^n = zeros(3, N);  % Spesifikk kraft
v^n = zeros(3, N);  % Hastighet
p^n = zeros(3, N);  % Posisjon
w^n = zeros(3, N);  % vinkelhastighet

```

Uttrykkene for bevegelse

$$p(t) = (t_i - t_{i+1}) = \frac{f}{\omega} \left[(t_{i+1} - \frac{1}{\omega} \sin(\omega t_{i+1})) - (t_i - \frac{1}{\omega} \sin(\omega t_i)) \right]$$

$$v(t) = v(t_i - t_{i+1}) = -\frac{f}{\omega} [\cos(\omega t_{i+1}) - \cos(\omega t_i)]$$

$$a(t) = a(t_i - t_{i+1}) = f [\sin(\omega t_{i+1}) - \sin(\omega t_i)]$$

$$f = \left(\frac{30}{2 - \frac{1}{\pi} \sin(\omega 2)} \right) \omega$$

$$\omega = \pi$$

for-løkker for generering av omløpet og pausering i hjørnene

```

for i = 1 : 200
    f^n(1, i) = a(i)           % Bevegelse i x-retning: 0 - 0,3 m
    v^n(1, i) = v(i)
    p^n(1, i) = p(i)
end

```

```

for i = t1 : t2
    p^n(i) = p^n(t1)          % Pausering i første hjørne
end

```

```

for i = 1 : 200
    f^n(2, i + t2) = a(i)      % Bevegelse i y-retning: 0 - 0,3 m
    v^n(2, i + t2) = v(i)
    p^n(2, i + t2) = v(i)
    p^n(1, i + t2) = p^n(1, t2)
end

```

```

for i = t3 : t4

```



```

     $\underline{p}^n(i) = \underline{p}^n(t_3)$                                 % Pausering i andre hjørne
end

for  $i = 1 : 200$ 
     $\underline{f}^n(1, i + t_4) = -a(i)$                                 % Bevegelse i x-retning: 0,3 - 0 m
     $\underline{v}^n(1, i + t_4) = -v(i)$ 
     $\underline{p}^n(1, i + t_4) = -p(i)$ 
     $\underline{p}^n(2, i + t_4) = -\underline{p}^n(2, t_4)$ 
end

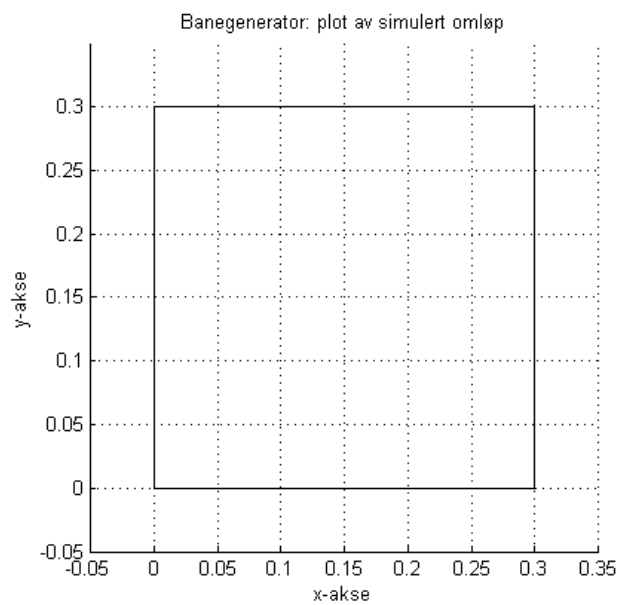
for  $i = t_5 : t_6$ 
     $\underline{p}^n(i) = \underline{p}^n(t_5)$                                 % Pausering i tredje hjørne
end

for  $i = 1 : 200$ 
     $\underline{f}^n(2, i + t_6) = -a(i)$                                 Bevegelse i y-retning: 0,3 - 0 m
     $\underline{v}^n(2, i + t_6) = -v(i)$ 
     $\underline{p}^n(2, i + t_6) = -p(i)$ 
end

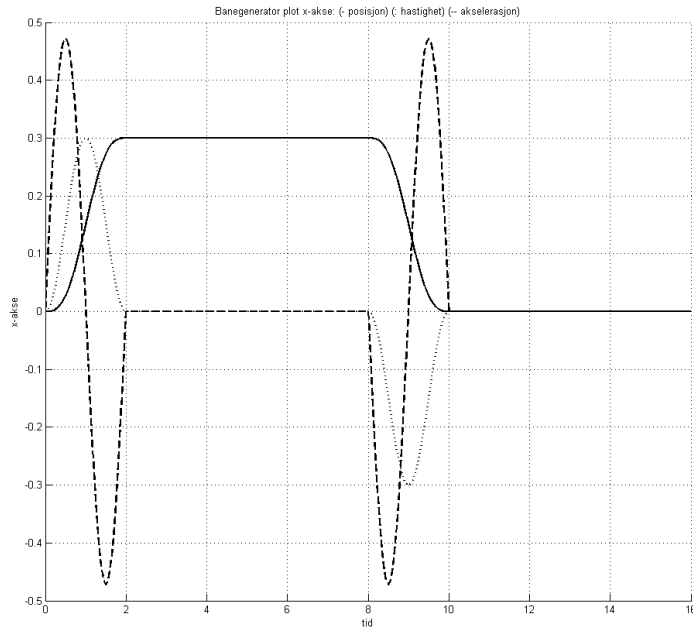
for  $t_7 : t_8$ 
     $\underline{p}^n(i) = \underline{p}^n(t_7)$                                 % Pausering i fjerde hjørne
end

```

Figur 5.2 viser det simulerte omløpet fra banegeneratoren. Figur 5.3 viser det simulerte av x i forhold til tiden. Heltrukken linje viser posisjon, prikket linje viser hastighet og streket linje (–) viser akselerasjon.



Figur 5.2: Banegenerator - simulert omløp



Figur 5.3: Banegenerator - simulert for x-akse

```


$$\underline{g}^n = [0 ; 0 ; -9.81]$$


$$\underline{f}^n = \underline{f}^n + \underline{g}^n$$


$$\underline{\omega}^n = \text{zeros}(3, N)$$

    % Legger på gravitasjon
    % vinkelhastighet=0

```

Legger på støy

Det er ukjent hvilke sensorer som er brukt i mobiltelefonen. Verdiene som her er brukt for å generere støy og feil er hentet fra [9] og [10].

Mobiltelefonen endrer ikke orientering under omløpet. Det gjør at $R_n^p = I$ og følgelig at $\underline{f}^p = \underline{f}^n$ og at $\underline{\omega}_p^{np} = \underline{\omega}_p^n$

```

VRW=0.0033          % Velocity Random Walk
ARW=0.0187          % Angular Random Walk

```

```


$$Q_a = \text{diag}([VRW, VRW, VRW])$$


$$Q_g = \text{diag}([ARW, ARW, ARW])$$


```

```


$$\underline{\beta}^a = 0.001962 \frac{m}{s^2}$$


$$\underline{\beta}^g = 0.01443 \frac{\text{deg}}{s}$$

    % Bias til akselerometret
    % Bias til gyrometret

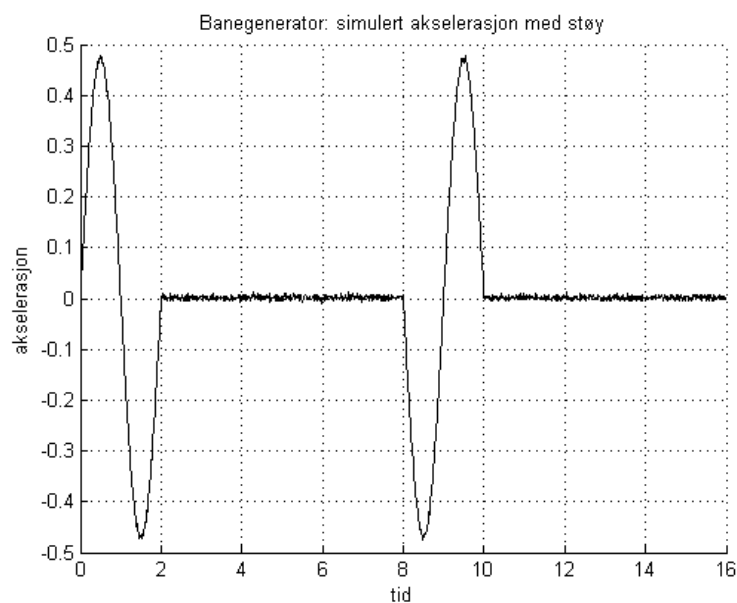
```

```

for
     $\tilde{f}^p(t_k) = \underline{f}^n + Q_a * randn(3,1) + \underline{\beta}^g$     % Trekker diskret hvit støy
     $\tilde{\omega}_p^{np}(t_k) = M_\omega + Q_g * randn(3,1) + \underline{\beta}^a$ 
end

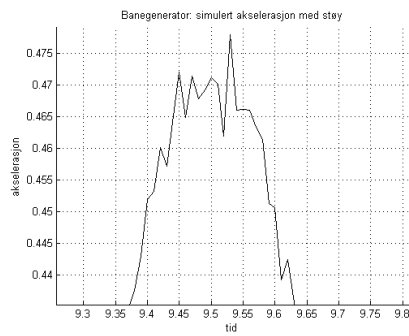
```

Figur 5.4 viser akselerasjon simulert med støy. Figurene 5.5 (a) og (b) viser utsnitt av samme simulering for bedre å få fram støyen i simuleringa.

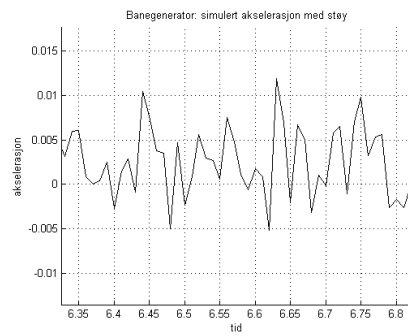


Figur 5.4: akselerasjon simulert med støy

Vinkelhastigheten var i banegeneratoren satt til 0. Etter simulering av nominell bane er det lagt på støy. Figur 5.6 (a) og (b) viser simulert vinkelhastighet henholdsvis uten og med støy.

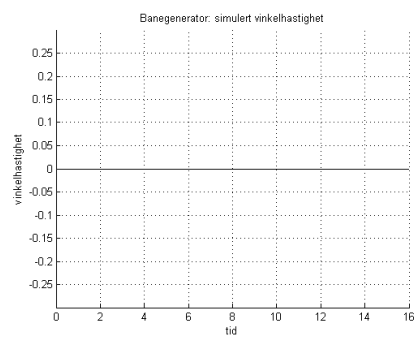


(a) mobiltelefonens akselerasjon er størst

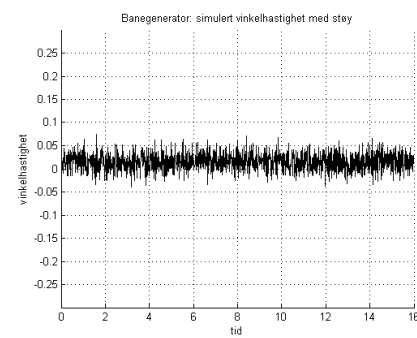


(b) mobiltelefonen er i ro

Figur 5.5: Utsnitt av akselerasjon med støy



(a) nominell vinkelhastighet er lik 0



(b) vinkelhastighet med støy

Figur 5.6: Vinkelhastighet uten og med støy

5.4 Simulering av navigasjonligninger

5.4.1 Integrasjonsrutine

Gitt den ordinære differensialligningen:

$$\dot{x} = f(t, x)$$

Eulers forover-metode er en integrasjonsrutine som blir brukt for å løse differensialligningen. En tilnærming til Eulers metode er linearisering gjennom Taylorrekke-utvikling av funksjonen f med tidssteg Ts

$$x(t_k) = \sum_{i=0}^{\infty} \frac{x}{t!} (x - a)^n$$

$$x(t_k) = x(t_{k-1}) + Ts\dot{x}(t_{k-1}) + \frac{1}{2}Ts^2\ddot{x}(t_{k-1}) + \dots$$

Eulers forovermetode inkluderer bare 1. ordens-elementene og det gir:

$$x_{k+1} = x_k + Ts * f(t_k, x_k)$$

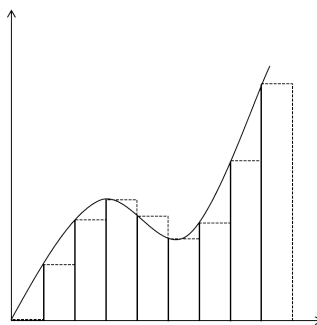
Med eulers forovermetode er tilnærmingen til den deriverte:

$$\dot{x}(k) \approx \frac{x(k+1) - x(k)}{Ts}$$

Eulers forovermetode kan skrives som :

$$y_{k+1} = y_k + Ts f(t_k, y_k)$$

Eulers metode antar konstant verdi i tidsintervallet. Som figur 5.7 viser gir integrasjonen en feil. Denne feilen kan reduseres ved å bruke et kortere tidsintervall.



Figur 5.7: Eulers forovermetode

5.4.2 Pseudokode for simulering av navigasjonsligninger

```

 $\underline{x} = [p ; v ; \theta]$ 
 $\tilde{\underline{x}} = \text{zeros}(9, N)$ 
 $\underline{u}(t_k) = [\tilde{f}^p(t_k) ; \tilde{\omega}_p^{np}(t_k)]$ 
 $R_p^n = \text{zeros}(3, 3, N)$ 

for i=1:N
     $\tilde{\underline{x}}_{k+1} = \tilde{\underline{x}}_k + \text{funksjon}(\tilde{\underline{x}}(k), \underline{u}(k), R_p^n(k))$ 
     $R_p^n(k+1) = R_p^n(k) + Ts * R_p^n(k) S(\underline{\omega}_k)$ 
end

function [ f ] = funksjon(x,u,R)
     $g = [0 ; 0 ; -9, 81]$ 
     $f(1 : 3, 1) = x(4 : 6)$ 
     $f(4 : 5, 1) = Ru(1 : 3) - g$ 
end

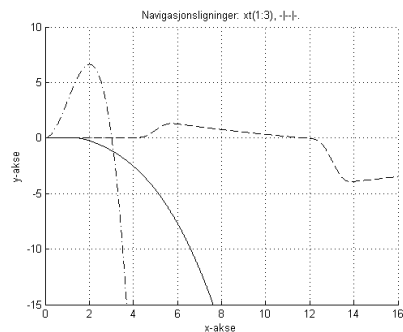
function [Rpn] = Rpn (θ)

$$R_p^n = \begin{bmatrix} 1 & \sin \theta_1 * \tan \theta_2 & \cos \theta_1 * \tan \theta_2 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \frac{\sin \theta_1}{\cos \theta_2} & \frac{\cos \theta_1}{\cos \theta_2} \end{bmatrix}$$

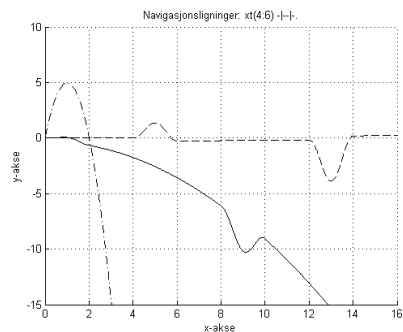
end

```

Figurene 5.8, 5.9 og 5.10 viser tre kjøringar av navigasjonsligningene. Resultatene spriker veldig og kurveformen virker ikke riktig. Simuleringene fra banegeneratoren er korrekte og det antas at det må være en feil i simuleringen av navigasjonsligningene.

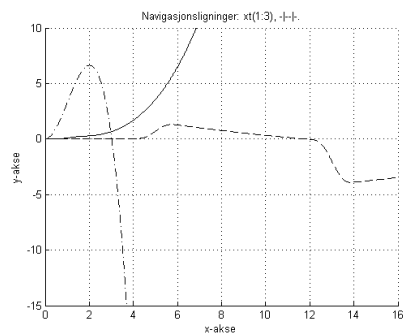


(a) $xt(1:3)$

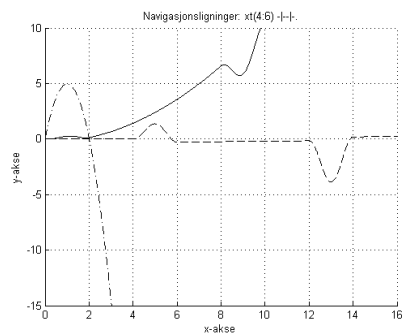


(b) $xt(4:6)$

Figur 5.8: Navigasjonsligninger simuleringseksempel 1

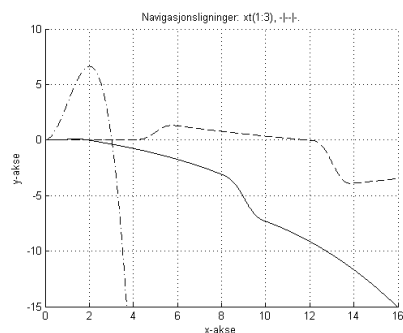


(a) $xt(1:3)$

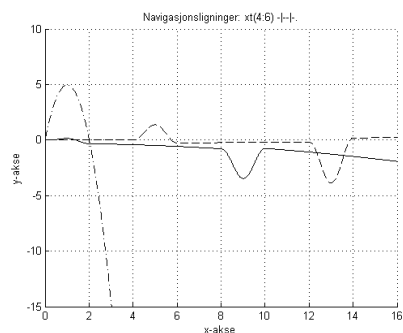


(b) $xt(4:6)$

Figur 5.9: Navigasjonsligninger simuleringseksempel 2



(a) $xt(1:3)$



(b) $xt(4:6)$

Figur 5.10: Navigasjonsligninger simuleringseksempel 3

6 Kalman-filter (KF)

(Kilder brukt i dette kapitlet er [1], [2], [3], [4], [5] og [7].)

Kalman-filtret (KF) er oppkalt etter Ungareren Rudolf Emil Kalman som var en av utviklerne bak KF.

Et KF er en estimator som opererer rekursivt og baserer seg på kunnskap om prosess og målinger. Kunnskapen om prosess og målinger modelleres inn i KFet i differensialligningene som beskriver systemet.

Sensorer gir målinger av systemets tilstand, men målinger vil alltid være påvirket av feil og støy. Et KF er en optimalisert estimator for å bestemme det beste estimatet av tilstanden og beregne et minimum variansestimert. KF kan brukes til å estimere tidligere, nåværende og framtidig tilstand.

Kalmanfilteret er altså en estimator. En estimator kan brukes der hvor direkte målinger av tilstander er umulig, dyrt eller på andre måter uhensiktsmessig.

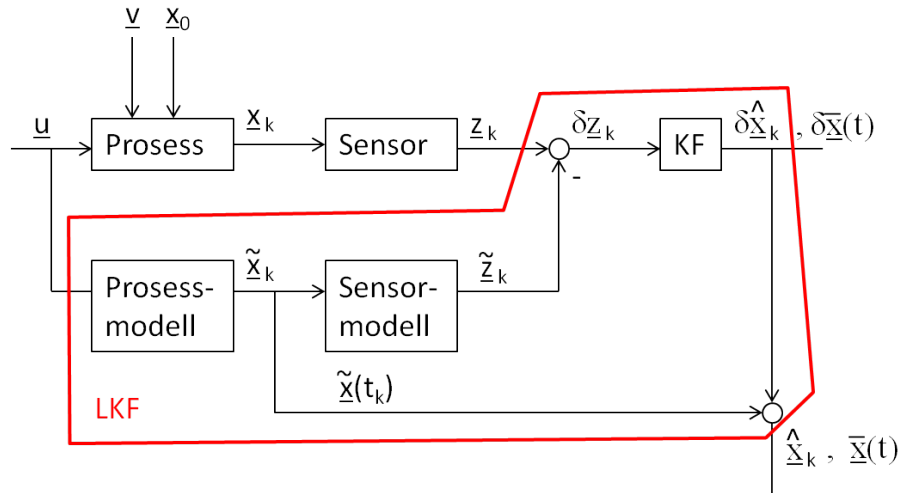
6.1 KF har to faser - prediksjon og måleoppdatering

I prediksjonsfasen (tidsoppdatering (TO)) beregner KFet estimatene for den sanne og ukjente tilstanden med tilhørende usikkerheter. Når neste sett med målinger kommer oppdateres estimatene (måleoppdatering (MO)). Estimaten oppdateres hvor de vektas med større tyngde for målinger med større sikkerhet. En måte å vekte målingene på er ved å bruke gain i tilknytning til målingene som uttrykker tiltroen til en måling. Med lav gain følger KFet modellprediksjonen nærmere, glatter ut støy og reduserer følsomheten. Med høyere gain følger KFet i større grad målingene og i mindre grad modellprediksjonen, da den blir mer følsom for endringer i målingene.

Et KF er rekursivt og kan brukes i sanntid basert på den siste målingen og den forrige estimerte tilstanden. For et KF forutsettes det at det underliggende systemet er et lineært dynamisk system og at alle feil og målinger er Gaussisk. Treghtsnavigasjon er ikke en lineær prosess, men KF kan tilpasses slik at det kan brukes i en ikke-lineær prosess. Noen av disse metodene kalles linearisert KF, unscented KF og utvidet KF. I denne oppgaven vil det bli brukt et linearisert KF.

6.2 Linearisert Kalman-filter (LKF)

Med et ulineært system kan LKF brukes der løsningen kan lineariseres rundt et arbeidspunkt. Et LKF har ingen tilbakekoblinger.



Figur 6.1: Blokkskjema av linearisert Kalman-filter

6.2.1 Ligninger for LKF

Tidsoppdatering (TO) - prediksjonsligningene

Mellom hver av målingene fra sensorene vil et KFet gi estimater for hva tilstanden vil være - denne fasen kalles tidsoppdatering (TO).

\bar{x}_k : prediktert (a priori) tilstandsvektor

\tilde{x} : nominell tilstandsvektor

\bar{P} : kovariansen til prediktert tilstandsvektor

G : prosesstøymatrisa

\tilde{Q} : prosesstøyens spektraltetthet

$\dot{\tilde{x}} = f(\tilde{x}, u)$, hvor $\tilde{x}(\hat{t}_k) = \tilde{x}(\bar{t}_k)$, og $t \in [\hat{t}_k, \bar{t}_{k+1}]$

$\delta \dot{\tilde{x}}(t) = F(t) \cdot \delta \tilde{x}(t)$; $\delta \tilde{x}(\hat{t}_k) = \delta \hat{x}_k$

$\bar{x}(t) = \tilde{x}(t) + \delta \tilde{x}(t)$

$\dot{\bar{P}}(t) = F(t)\bar{P}(t) + \bar{P}(t)F^T(t) + G(t)\tilde{Q}(t)G^T(t)$, $\bar{P}(\hat{t}_k) = \hat{P}_k$ og \hat{P}_0 gitt

Måleoppdatering (MO)

Estimatene blir oppdatert når neste måling kommer inn.

\hat{x}_k : estimert (a posteriori) tilstandsvektor
 K : KF-forsterkninga
 H : Målematrisa
 \hat{P}_K : estimert kovarians for tilstandsvektor
 z : måling

$$\begin{aligned}
 \delta z_k &= z_k - \tilde{z}_k \\
 \delta \hat{x}_k &= \delta \bar{x}_k + K_k(\delta z_k - H_k \delta \bar{x}_k) \\
 \hat{x}_k &= \bar{x}_k + \delta \hat{x}_k \\
 K_k &= \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}, \bar{P}_k = \bar{P}(t_k) \\
 \hat{P}_k &= (I - K_k H_k) \bar{P}_k, \hat{P}_0 \text{ gitt}
 \end{aligned}$$

Transformasjonfeilen referert n-systemet

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^n - \underline{f}^b$	$\delta \underline{x}^n = \underline{x}^n - \tilde{x}^n$	$R_b^n = R(\underline{\varepsilon}) \tilde{R}_b^n$
$\delta \underline{\omega} = \underline{\omega}_{nb}^b - \tilde{\omega}_{nb}^b$	$\delta \underline{v}^n = \underline{v}^n - \tilde{v}^n$	$R(\underline{\varepsilon}) = I + S(\underline{\varepsilon})$

Fysisk system	Navigasjonsligninger	Feilligninger
$\dot{\underline{x}}^n = \underline{v}^n$	$\dot{\tilde{x}}^n = \tilde{v}^n$	$\delta \dot{\underline{x}} = \delta \underline{v}$
$\dot{\underline{v}}^n = R_b^n \underline{f}^n - \underline{g}^n$	$\dot{\tilde{v}}^n = \tilde{R}_b^n \tilde{f}^b - \underline{g}^n$	$\delta \dot{\underline{v}}^n = -S(\tilde{R}_b^n \tilde{f}^b)_{\underline{\varepsilon}} + \tilde{R}_b^n \delta \underline{f}$
$\dot{\underline{R}}_b^n = R_b^n S(\underline{\omega}_{nb}^b)$	$\dot{\tilde{R}}_b^n = \tilde{R}_b^n S(\tilde{\omega}_{nb}^b)$	$\dot{\underline{\varepsilon}} = \tilde{R}_b^n \delta \underline{\omega}$

Dette gir alle formlene for å utvikle feilligningene for KF.

6.2.2 Diskretisering

Diskretisering bringer kontinuerlige prosesser over på en diskretisert form. Det er nødvendig, da en datamaskin kun håndterer diskrete verdier.

Har den kontinuerlige tilstandsmodellen:

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + L(t)\underline{u}(t) + G(t)\underline{v}(t)$$

Den kan diskretiseres som et integral over et tidsrom. Gjennom tidsrommet anses G og L som konstante.

$$x_{k+1} = \underbrace{e^{F((k+1)-k)}}_{\Phi_k x_k} + \underbrace{\int_k^{k+1} e^{F((k+1)-\tau)} L_k \underline{u}(\tau) d\tau}_{\Lambda_k \underline{u}_k} + \underbrace{\int_k^{k+1} e^{F((k+1)-\tau)} G_k \underline{v}(\tau) d\tau}_{\Gamma_k \underline{v}_k}$$

hvor:

$$\underline{x}_{k+1} = \Phi_k \underline{x}_k + \Lambda_k \underline{u}_k + \Gamma_k \underline{v}_k$$

Pådraget vil være 0 og pådragsmatrisa faller bort.

$$\bar{P}_{k+1} = \Phi_k \bar{P}_k \Phi_k^T + S, \text{ hvor } S = \Gamma Q \Gamma^T = \Gamma \tilde{Q} \Gamma^T$$

$$S = \Gamma_k Q \Gamma_k^T = \int_k^{k+1} \Phi_k G_k \tilde{Q} G_k^T \Phi_k^T \approx G_k \tilde{Q} G_k^T \Delta_t$$

Ved bruk av cholesky-faktorisering kan Γ og Q spaltes:

$$S = U D U^T$$

$$\text{chol}(S)'$$

6.2.3 Pseudokode for Kalman-filter

$d2r = \pi/180$	% Omgjøring fra grader til radianer
$\delta p = 0.001$	% Usikkerhet i posisjon (mm)
$\delta v = 0.001$	% Usikkerhet i hastighet (m/s)
$\delta \varepsilon = 1 * d2r$	% Usikkerhet i orientering
$\delta \beta^a$	% Bias for akselerometer
$\delta \beta^g$	% Bias for gyrometer

Kalman-filteret har en ytre løkke som går alle tidsstegene. For hver tidsoppdatering (TO) trekkes programmet inn i en egen løkke for TO. Tilsvarende er det for måleoppdatering (MO).

for $t = t_0 : N$

MO gjøres bare én gang ved begynnelsen av pauseringen i hvert hjørne. Tidssteget er på 1 sekund og er angitt som Δ_k .

MO - måleoppdatering

if $((t == t_0) \mid (t == t_1) \mid (t == t_3) \mid (t == t_5) \mid (t == t_7))$

for $k = 1 : 1$ % $\Delta_t = 1s$

$$H_k = [0, I, 0]$$

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}$$

$$\hat{P}_k = (I - K_k H_k) \bar{P}_k$$

$$\underline{z}_k = H_k * \delta \underline{x}$$

$$\delta \underline{z}_k = \underline{z}_k - \tilde{\underline{z}}_k$$

$$\delta \hat{\underline{x}}_k = \delta \bar{\underline{x}}_k + K_k (\delta \underline{z}_k - H_k \delta \bar{\underline{x}}_k)$$

$$\hat{\underline{x}}_k = \tilde{\underline{x}}_k + \delta \hat{\underline{x}}_k$$

end % end for

end % end MO

TO gjøres fra og med begynnelsen av hver bevegelse og i tidssteg på 0,01 sekund. Funksjonen $k2dS$ er vist under KFet. Tidssteget på 0,01 sekund er konstant under intervallet og er angitt som Δ_l . Det er derfor satt en subscript d på prosessstøyens kovarians (Q_d) etter Q for å tydeliggjøre dette. I og med

at Q_d er konstant under intervallet blir verdien kvadrert.

TO - tidsoppdatering

if $((t == t_0) \mid (t == t_2) \mid (t == t_4) \mid (t == t_6))$

for $l = 0.01 : 0.01 : 1.00$ % $\Delta_t = 0.01s$

$$F = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & -S(\tilde{R}_b^n \tilde{f}^b) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 0 \\ \tilde{R}_b^n & 0 \\ 0 & \tilde{R}_b^n \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Phi = e^{(F * T_s)} = I + \Delta_l F(t_k)$$

$$S = k2dS(F, G, Q, T_s) \quad \% = GQ_dG^T \Delta_l^2$$

$$\Gamma = chol(S)'$$

$$\delta \bar{x}_{l+1} = \Phi \delta \bar{x}_l$$

$$\bar{P}_{l+1} = \Phi \bar{P}_l \Phi^T + GQ_dG^T \Delta_l^2$$

$$\bar{x}_l = \tilde{x}_l + \delta \bar{x}_l$$

$$\bar{R}_b^n = (I + S(\varepsilon)) \tilde{R}_b^n$$

end % end for

end % end TO

end % end Kalmanfilter

function [S] = k2dS (F,G,Q_d,G^T,Δ_l²)

$$A = \begin{bmatrix} F & GQ_dG^T \\ 0 & -F^T \end{bmatrix}$$

$$S = \frac{\Phi_{11}}{\Phi_{22}}$$

end % end function

7 Konklusjon

Det er laget pseudokode for banegenerator, simulering av navigasjonligninger og Kalman-filter. Kalman-filteret har løpende tidsoppdatering og måleoppdateres ved hver pausering i et hjørne. Resultatet fra banegeneratoren viser at simulerte sensordata er korrekte. I og med at det er ukjent hvilke akselerometer og gyrometer som er brukt i mobiltelefonen vil verdiene som er brukt for feil og støy gi usikkerhet i forhold til hvor godt de passer med hva som er reelt for sensorene.

8 Videre arbeid

Målet er å kunne måle et objekt ved å flytte mobiltelefonen fra hjørne til hjørne på en eske.

Ut fra pseudokoden må det lages et fullstendig Kalman-filter. Det er også antagelig en feil i simuleringen av navigasjonsligningene som må rettes. Videre må det lages et program som leser inn loggede rå sensordata, kjører dette gjennom Kalman-filtret og plotter bevegelsesforløpet. Viss det viser seg at navigasjonssystemet blir nøyaktig nok må det avslutningsvis lages et program som beregner areal eller volum for det målte objektet.

Nøyaktigheten til navigasjonssystemet kan tilnærmes ved å analysere resultatet fra kjøring med loggede sensordata fra et kjent bevegelsesmønster.

I og med at det er ukjent hvilke sensorer som er integrert i mobiltelefonen vil det være nyttig å lage et program som analyserer karakteristikk til sensorene. Verdiene som er brukt for feil og støy er hentet fra masteroppgavene [9] og [10]. Når tilstrekkelig med analyseinformasjon foreligger må det vurderes om verdiene skal endres.

Fra en mobiltelefon som ligger i ro skal utgangsverdien fra sensorene være 0. På sensorsignalet vil det alltid være støy og feil. Ved å analysere måledata fra en slik situasjon kan det estimeres verdier for biasfeil ved å middle signalet over tid.

For gyrometret skal hvit støy i gjennomsnitt være 0. Viss signalet integreres for å finne vinkelen skal resultatet også optimalt være 0. Den hvite støyen vil gjøre at integrasjonen vil hoppe rundt. For en mobiltelefon i ro kan det analyseres hvordan integrasjonsresultatet blir ved å kjøre flere tester hvor mobiltelefonen står i ro. Tilsvarende vil gjelde for akselerometeret.

De fleste (om ikke alle) smarttelefoner har integrerte temperatursensorer. Det kan være interessant å simulere sensorkarakteristikk under forskjellige temperaturforhold for å se om sensorutslagene påvirkes. Kanskje kan logging fra smarttelefonens integrerte temperatursensor brukes i en slik sammenheng.

9 Kilder og referanser

- [1] O. Hallingstad. Forelesningsnotater: Stokastiske systemer, UNIK 4500, 2010.
- [2] O. Hallingstad. Forelesningsnotater: Matematisk modellering av dynamiske systemer, UNIK 4550, 2011.
- [3] O. Hallingstad. Matematisk modellering av dynamiske systemer, notat UNIK, 2008.
- [4] O. Hallingstad. Eksempler på TNS-modeller, notat UNIK, 2004.
- [5] O. Hallingstad. Sammendrag av modeller og ligninger for Kalman-filteret anvendt på et ulineært system, notat UNIK.
- [6] www.no.wikipedia.org, april 2012.
- [7] www.en.wikipedia.org, april 2012.
- [8] www.mathworld.wolfram.com/, april 2012.
- [9] Nerholm, P. A. Masteroppgave, 2011
- [10] Grahn, J. S. Masteroppgave, 2011
- [11] Hamra, S. Masteroppgave, 2011

Vedlegg A

O. Hallingstad, notater

- O. Hallingstad, Sammendrag av modeller og ligninger for Kalman-filteret anvendt på et ulineært system
- O. Hallingstad, Eksempler på TNS-modeller, 2004

	SAMMENDRAG AV MODELLER OG LIKNINGER FOR KALMANFILTERET ANVENT PÅ ET ULINEÆRT SYSTEM			
S^{US}	Sann ulineær systemmodell	$\hat{x} = \underline{f}(\underline{x}, \underline{u}) + G\underline{v}$ $\underline{z}_k = \underline{h}_k(\underline{x}_k) + \underline{w}_k$	$\underline{x}_0 \sim \mathcal{N}(\hat{\underline{x}}_0, \hat{P}_0), \underline{v}_k \sim \mathcal{N}(0, Q\delta(t - \tau))$ $\underline{w}_k \sim \mathcal{N}(0, R\delta_{kl})$	$E\{\underline{x}_0 \underline{w}_k^T\} = 0, E\{\underline{x}_0 \underline{v}^T(t)\} = 0$ $E\{\underline{v}(t) \underline{w}_k^T\} = 0$
\mathcal{F}^{UD}	Ulineær det. filtermodell	$\hat{\underline{x}} = \underline{f}^*(\hat{\underline{x}}, \underline{u})$ $\hat{\underline{z}}_k = \underline{h}_k^*(\hat{\underline{x}}_k)$	$\hat{\underline{x}}_0 = T\hat{\underline{x}}_0$ $T = \begin{bmatrix} I & 0 \end{bmatrix}$	
S^{LS}	Sann lineær feilmodell	$\delta \hat{\underline{x}} = F(\hat{\underline{x}}, t) \delta \underline{x} + G\underline{v}$ $\delta \underline{z}_k = H_k(\hat{\underline{x}}) + \underline{w}_k$	$\delta \underline{x}_0 \sim \mathcal{N}(\hat{\underline{x}}_0 - T^T \hat{\underline{x}}_0, \hat{P}_0), \underline{v} \sim \mathcal{N}(0, Q\delta(t - \tau))$ $\underline{w}_k \sim \mathcal{N}(0, R\delta_{kl})$	$E\{\delta \underline{x}_0 \cdot \underline{w}_k^T\} = 0, E\{\delta \underline{x}_0 \cdot \underline{v}^T(t)\} = 0$ $E\{\underline{v}(t) \cdot \underline{w}_k^T\} = 0$
\mathcal{F}^{LS}	Lineær filtermodel	$\delta \hat{\underline{x}}^* = F^*(\hat{\underline{x}}, t) + G^* \underline{v}^*$ $\delta \underline{z}_k = H^*(\hat{\underline{x}}) + \underline{w}^*$	$\delta \underline{x}_0^* \sim \mathcal{N}(\hat{\underline{x}}_0^*, \hat{P}_0^*), \underline{v}^* \sim \mathcal{N}(0, Q^*\delta(t - \tau))$ $\underline{w}_k^* \sim \mathcal{N}(0, R^*\delta_{kl})$	$E\{\underline{x}_0^* \cdot \underline{w}_k^{*T}\} = 0, E\{\underline{x}_0^* \cdot \underline{v}^{*T}(t)\} = 0$ $E\{\underline{v}^*(t) \cdot \underline{w}_k^{*T}\} = 0$
	Dimensjoner	$\dim \underline{x} = \dim \delta \underline{x} = n_x \geq \dim \hat{\underline{x}} =$	$\dim \delta \underline{x}^* = n_{x^*}$	
	Filter:	TKF-Tilbakekoblet Kalmanfilter	LKF-Linearisert Kalmanfilter	UKF-Utvidet Kalmanfilter
IV	Initialv. $t = \hat{t}_0^+$	$\hat{\underline{x}}(\hat{t}_0^+) = \hat{\underline{x}}_0, \delta \hat{\underline{x}}_0 = 0, P_0 : \text{gitt}$	$\hat{\underline{x}}(\hat{t}_0^+) = \hat{\underline{x}}_0, \delta \hat{\underline{x}}_0 = 0, P_0 : \text{gitt}$	$\hat{\underline{x}}_0 = \hat{\underline{x}}(\hat{t}_0^+) = \hat{\underline{x}}_0, P_0 : \text{gitt}$
MO	Måleoppdatering	$\delta \underline{z}_k = \underline{z}_k - \hat{\underline{z}}_k$ $\delta \hat{\underline{x}}_k = \delta \hat{\underline{x}}_k + K_k^*(\delta \underline{z}_k - H_k^* \delta \hat{\underline{x}}_k)$	$\delta \underline{z}_k = \underline{z}_k - \hat{\underline{z}}_k$ $\delta \hat{\underline{x}}_k = \delta \hat{\underline{x}}_k + K_k^*(\delta \underline{z}_k - H_k^* \delta \hat{\underline{x}}_k)$ $\hat{\underline{x}}_k = \hat{\underline{x}}_k + \delta \hat{\underline{x}}_k$	$\hat{\underline{x}}_k = \hat{\underline{x}}_k + K_k^*(\underline{z}_k - \hat{\underline{z}}_k^*(\hat{\underline{x}}_k))$
TK	Tilbakekobl. ved $t = \hat{t}_k^+$	$K_k^* = \hat{P}_k H_k^{*T} (H_k \hat{P}_k H_k^{*T} + R_k^*)^{-1}$ $\hat{\underline{x}}_k^+ = \hat{\underline{x}}_k + S^* \delta \hat{\underline{x}}_k$ $\delta \hat{\underline{x}}_k^+ = (I - S^*) \delta \hat{\underline{x}}_k$ $\hat{P}_k^+ = \hat{P}_k$	$\hat{P}_k = (I - K_k^* H_k^*) \hat{P}_k$	
TO	Tidsoppdatering	$\hat{\underline{x}} = \underline{f}^*(\hat{\underline{x}}, \underline{u}), \hat{\underline{x}}(\hat{t}_k^+) = \hat{\underline{x}}_k^+, t \in [\hat{t}_k^+, \hat{t}_{k+1}]$ $\delta \hat{\underline{x}}(t) = F^*(t) \cdot \delta \hat{\underline{x}}(t); \delta \hat{\underline{x}}(\hat{t}_k^+) = \delta \hat{\underline{x}}_k^+$ $\hat{\underline{x}}(t) = \hat{\underline{x}}(t) + \delta \hat{\underline{x}}(t)$	$\hat{\underline{x}} = \underline{f}^*(\hat{\underline{x}}, \underline{u}), \hat{\underline{x}}(\hat{t}_k) = \hat{\underline{x}}(\hat{t}_k), t \in [\hat{t}_k, \hat{t}_{k+1}]$ $\delta \hat{\underline{x}}(t) = F^*(t) \cdot \delta \hat{\underline{x}}(t); \delta \hat{\underline{x}}(\hat{t}_k) = \delta \hat{\underline{x}}_k$ $\hat{\underline{x}}(t) = \hat{\underline{x}}(t) + \delta \hat{\underline{x}}(t)$	$\hat{\underline{x}} = \underline{f}^*(\hat{\underline{x}}, \underline{u}), \hat{\underline{x}}(\hat{t}_k) = \hat{\underline{x}}_k, t \in [\hat{t}_k, \hat{t}_{k+1}]$
		$\hat{P}(t) = F^*(t) \hat{P}(t) + \hat{P}(t) F^{*T}(t) + G^*(t)$ LKF ($S^* = 0$) og UKF ($S^* = I$) er	$\hat{Q}^*(t) G^{*T}(t); \hat{P}(\hat{t}_k) = \hat{P}_k$ spesialtilfeller av TKF	

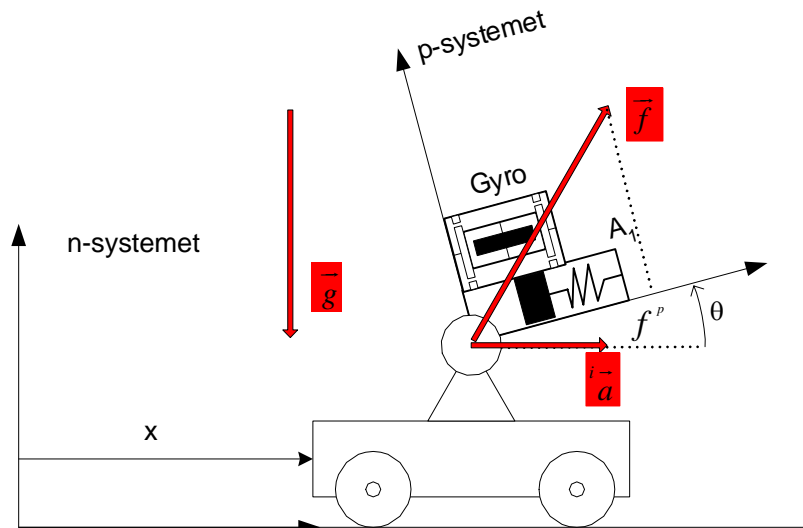
Eksempler på TNS-modeller

Oddvar Hallingstad

3. mars 2004

1 En-akset plattform

Figuren nedenfor viser en en-akset plattform (den måler akselerasjonen bare langs en akse) som kan dreie seg om en akse vinkelrett på papirplanet. Denne dreiningen måles av en gyro.



Figur 1: Enakset plattform

Feildefinisjoner		
$\delta f = f - \tilde{f}$	$\delta x = x - \tilde{x}$	$\delta \theta = \theta - \tilde{\theta}$
$\delta \omega = \omega - \tilde{\omega}$	$\delta v = v - \tilde{v}$	

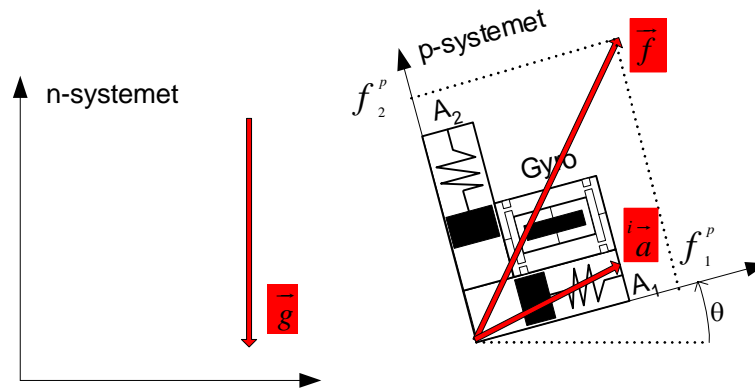
Legg merke til definisjonene av feil. De er valgt på denne måten fordi dette gir en standard systemmodell for et eventuelt Kalmanfilter. Målingene \tilde{f} og $\tilde{\omega}$ kan ses på som pådrag. Dersom vi måler posisjonen for bruk i et Kalmanfilter som måling må feilmodellen være $y = \tilde{x} = x + w$ hvor w er hvit støy.

Fysisk system	Mekanisert system	Feillikninger
$\dot{x} = v$	$\dot{\tilde{x}} = \tilde{v}$	$\delta \dot{x} = \delta v$
$\dot{v} = \frac{f - g \sin \theta}{\cos \theta}$	$\dot{\tilde{v}} = \frac{\tilde{f} - g \sin \tilde{\theta}}{\cos \tilde{\theta}}$	$\delta \dot{v} = \frac{1}{\cos^2 \tilde{\theta}} (\tilde{f} \sin \tilde{\theta} - g) \delta \theta + \frac{1}{\cos \tilde{\theta}} \delta f$
$\dot{\theta} = \omega$	$\dot{\tilde{\theta}} = \tilde{\omega}$	$\delta \dot{\theta} = \delta \omega$

Bare likningen for $\delta \dot{v}$ krever en egen utledning:

$$\begin{aligned} \delta \dot{v} &= \frac{f - g \sin \theta}{\cos \theta} - \frac{\tilde{f} - g \sin \tilde{\theta}}{\cos \tilde{\theta}} = \frac{\tilde{f} + \delta f - g \sin(\tilde{\theta} + \delta \theta)}{\cos(\tilde{\theta} + \delta \theta)} - \frac{\tilde{f} - g \sin \tilde{\theta}}{\cos \tilde{\theta}} = \frac{\delta f}{\cos \tilde{\theta}} + \left(\frac{\tilde{f} \sin \tilde{\theta}}{\cos^2 \tilde{\theta}} - g (1 + \tan^2 \theta) \right) \delta \theta + O(\delta \theta^2, \delta f^2) \\ &= \frac{1}{\cos^2 \tilde{\theta}} (\tilde{f} \sin \tilde{\theta} - g) \delta \theta + \frac{1}{\cos \tilde{\theta}} \delta f + O(\delta \theta^2, \delta f^2) \end{aligned}$$

2 To-akset plattform



Figur 2: To-akset plattform

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^p - \tilde{\underline{f}}^p$	$\delta \underline{x} = \underline{x}^n - \tilde{\underline{x}}^n$	$\delta \theta = \theta - \tilde{\theta}$
$\delta \omega = \omega - \tilde{\omega}$	$\delta \underline{v} = \underline{v}^n - \tilde{\underline{v}}^n$	$R_p^n = R(\tilde{\theta}) \left(I + \delta \theta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right)$

Den elementære rotasjonsmatrisa R er gitt ved

$$R_p^n = R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Feillikningene skrevet på vektorform:

Fysisk system	Mekanisert system	Feillikninger
$\dot{\underline{x}}^n = \underline{v}^n$	$\dot{\tilde{\underline{x}}}^n = \tilde{\underline{v}}^n$	$\delta \dot{\underline{x}} = \delta \underline{v}$
$\dot{\underline{v}}^n = R_p^n \underline{f}^p - \underline{g}^n$	$\dot{\tilde{\underline{v}}}^n = \tilde{R}_p^n \tilde{\underline{f}}^p - \underline{g}^n$	$\delta \dot{\underline{v}}^n = R_3(\tilde{\theta}) \begin{bmatrix} -\tilde{f}_y \\ \tilde{f}_x \end{bmatrix} \delta \theta + R_3(\tilde{\theta}) \delta \underline{f}$
$\dot{\theta} = \omega$	$\dot{\tilde{\theta}} = \tilde{\omega}$	$\delta \dot{\theta} = \delta \omega$

Feillikningene skrevet på komponentform:

Fysisk system	Mekanisert system	Feillikninger
$\dot{\underline{x}} = \underline{v}$	$\dot{\tilde{\underline{x}}} = \tilde{\underline{v}}$	$\delta \dot{\underline{x}} = \delta \underline{v}$
$\dot{v}_1 = f_x \cos \theta - f_y \sin \theta$	$\dot{\tilde{v}}_1 = \tilde{f}_x \cos \tilde{\theta} - \tilde{f}_y \sin \tilde{\theta}$	$\delta \dot{v}_1 = -(\tilde{f}_x \sin \tilde{\theta} + \tilde{f}_y \cos \tilde{\theta}) \delta \theta + \delta f_x \cos \tilde{\theta} - \delta f_y \sin \tilde{\theta}$
$\dot{v}_2 = f_x \sin \theta + f_y \cos \theta - g$	$\dot{\tilde{v}}_2 = \tilde{f}_x \sin \tilde{\theta} + \tilde{f}_y \cos \tilde{\theta} - g$	$\delta \dot{v}_2 = (\tilde{f}_x \cos \tilde{\theta} - \tilde{f}_y \sin \tilde{\theta}) \delta \theta + \delta f_x \sin \tilde{\theta} + \delta f_y \cos \tilde{\theta}$
$\dot{\theta} = \omega$	$\dot{\tilde{\theta}} = \tilde{\omega}$	$\delta \dot{\theta} = \delta \omega$

Utleddning av likningen for hastighetsfeil:

$$\delta \dot{\underline{v}}^n = R_p^n \underline{f}^p - \tilde{R}_p^n \tilde{\underline{f}}^p = R(\tilde{\theta}) \left(I + \delta \theta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right) (\tilde{\underline{f}}^p + \delta \underline{f}) - R(\tilde{\theta}) \tilde{\underline{f}}^p = R(\tilde{\theta}) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tilde{\underline{f}}^p \delta \theta + R(\tilde{\theta}) \delta \underline{f} = R(\tilde{\theta}) \begin{bmatrix} -\tilde{f}_y \\ \tilde{f}_x \end{bmatrix} \delta \theta + R(\tilde{\theta}) \delta \underline{f}$$

3 Tre-akset plattform, flat ikke-roterende jord

For en tre-akset plattform kan feilen defineres på flere måter.

3.1 Transformasjonsfeilen referert eulervinklene

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^p - \tilde{\underline{f}}^p$	$\delta \underline{x}^n = \underline{x}^n - \tilde{\underline{x}}^n$	$\delta \underline{\theta} = \underline{\theta} - \tilde{\underline{\theta}}$
$\delta \underline{\omega} = \underline{\omega}_{np}^p - \tilde{\underline{\omega}}_{np}^p$	$\delta \underline{v}^n = \underline{v}^n - \tilde{\underline{v}}^n$	

Jeg har et eget notat for utledningen her.

3.2 Transformasjonsfeilen referert n-systemet

Vi skal her se på feillikningene når feilen i beregningen av rotasjonsmatrisa R_p^n refereres til n-systemet.

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^p - \tilde{\underline{f}}^p$	$\delta \underline{x}^n = \underline{x}^n - \tilde{\underline{x}}^n$	$R_p^n = R(\underline{\varepsilon}) \tilde{R}_p^n$
$\delta \underline{\omega} = \underline{\omega}_{np}^p - \tilde{\underline{\omega}}_{np}^p$	$\delta \underline{v}^n = \underline{v}^n - \tilde{\underline{v}}^n$	$R(\underline{\varepsilon}) = I + S(\underline{\varepsilon})$

Fysisk system	Mekanisert system	Feillikninger
$\dot{\underline{x}}^n = \underline{v}^n$	$\dot{\tilde{\underline{x}}}^n = \tilde{\underline{v}}^n$	$\delta \dot{\underline{x}} = \delta \underline{v}$
$\dot{\underline{v}}^n = R_p^n \underline{f}^p - \underline{g}^n$	$\dot{\tilde{\underline{v}}}^n = \tilde{R}_p^n \tilde{\underline{f}}^p - \underline{g}^n$	$\delta \dot{\underline{v}}^n = -S(\tilde{R}_p^n \tilde{\underline{f}}^p) \underline{\varepsilon} + \tilde{R}_p^n \delta \underline{f}$
$\dot{R}_p^n = R_p^n S(\underline{\omega}_{np}^p)$	$\dot{\tilde{R}}_p^n = \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p)$	$\dot{\underline{\varepsilon}} = \tilde{R}_p^n \delta \underline{\omega}$

Bevis av d.l. for $\delta \underline{v}^n$:

$$\delta \dot{\underline{v}}^n = R_p^n \underline{f}^p - \underline{g}^n - (\tilde{R}_p^n \tilde{\underline{f}}^p - \underline{g}^n) = (I + S(\underline{\varepsilon})) \tilde{R}_p^n (\tilde{\underline{f}}^p + \delta \underline{f}) - \tilde{R}_p^n \tilde{\underline{f}}^p = S(\underline{\varepsilon}) \tilde{R}_p^n \tilde{\underline{f}}^p + \tilde{R}_p^n \delta \underline{f}$$

$$=$$

Bevis av d.l. for $\underline{\varepsilon}$:

$$\text{V. siden: } \dot{R}_p^n - \tilde{R}_p^n = S(\dot{\underline{\varepsilon}}) \tilde{R}_p^n + (I + S(\underline{\varepsilon})) \dot{\tilde{R}}_p^n - \dot{\tilde{R}}_p^n = S(\dot{\underline{\varepsilon}}) \tilde{R}_p^n + S(\underline{\varepsilon}) \dot{\tilde{R}}_p^n = S(\dot{\underline{\varepsilon}}) \tilde{R}_p^n + S(\underline{\varepsilon}) \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p)$$

$$\text{H. siden: } R_p^n S(\underline{\omega}_{np}^p) - \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p) = (I + S(\underline{\varepsilon})) \tilde{R}_p^n (S(\tilde{\underline{\omega}}_{np}^p) + S(\delta \underline{\omega})) - \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p) = S(\underline{\varepsilon}) \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p) + \tilde{R}_p^n S(\delta \underline{\omega})$$

Satt sammen:

$$S(\dot{\underline{\varepsilon}}) \tilde{R}_p^n + S(\underline{\varepsilon}) \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p) = S(\underline{\varepsilon}) \tilde{R}_p^n S(\tilde{\underline{\omega}}_{np}^p) + \tilde{R}_p^n S(\delta \underline{\omega})$$

$$S(\dot{\underline{\varepsilon}}) \tilde{R}_p^n = \tilde{R}_p^n S(\delta \underline{\omega})$$

$$S(\dot{\underline{\varepsilon}}) = \tilde{R}_p^n S(\delta \underline{\omega}) \tilde{R}_p^n \Leftrightarrow \dot{\underline{\varepsilon}} = \tilde{R}_p^n \delta \underline{\omega}$$

3.3 Transformasjonsfeilen referert p-systemet

Vi skal her se på feillikningene når feilen i beregningen av rotasjonsmatrisa R_p^n refereres til p-systemet.

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^p - \tilde{\underline{f}}^p$	$\delta \underline{x}^n = \underline{x}^n - \tilde{\underline{x}}^n$	$R_p^n = \tilde{R}_p^n R(\underline{\varepsilon})$
$\delta \underline{\omega} = \underline{\omega}_{np}^p - \tilde{\underline{\omega}}_{np}^p$	$\delta \underline{v}^n = \underline{v}^n - \tilde{\underline{v}}^n$	$R(\underline{\varepsilon}) = I + S(\underline{\varepsilon})$

Fysisk system	Mekanisert system	Feillikninger
$\dot{\underline{x}}^n = \underline{v}^n$	$\dot{\underline{\tilde{x}}}^n = \underline{\tilde{v}}^n$	$\delta \dot{\underline{x}} = \delta \underline{v}$
$\underline{\dot{v}}^n = \underline{R}_p^n \underline{f}^p - \underline{g}^n$	$\underline{\dot{\tilde{v}}}^n = \underline{\tilde{R}}_p^n \underline{\tilde{f}}^p - \underline{g}^n$	$\delta \underline{\dot{v}}^n = -\underline{\tilde{R}}_p^n S(\underline{\tilde{f}}^p) \underline{\varepsilon} + \underline{\tilde{R}}_p^n \delta \underline{f}$
$\dot{\underline{R}}_p^n = \underline{R}_p^n S(\underline{\omega}_{np}^p)$	$\dot{\underline{\tilde{R}}}_p^n = \underline{\tilde{R}}_p^n S(\underline{\omega}_{np}^p)$	$\dot{\underline{\varepsilon}} = -S(\underline{\omega}) \underline{\varepsilon} + \delta \underline{\omega}$

Bevis av d.l. for $\delta \underline{v}^n$:

$$\begin{aligned} \delta \underline{\dot{v}}^n &= \underline{R}_p^n \underline{f}^p - \underline{g}^n - \left(\underline{\tilde{R}}_p^n \underline{\tilde{f}}^p - \underline{g}^n \right) = \underline{\tilde{R}}_p^n (I + S(\underline{\varepsilon})) (\underline{\tilde{f}}^p + \delta \underline{f}) - \underline{\tilde{R}}_p^n \underline{\tilde{f}}^p = \underline{\tilde{R}}_p^n S(\underline{\varepsilon}) \underline{\tilde{f}}^p + \underline{\tilde{R}}_p^n \delta \underline{f} \\ &= -\underline{\tilde{R}}_p^n S(\underline{\tilde{f}}^p) \underline{\varepsilon} + \underline{\tilde{R}}_p^n \delta \underline{f} \end{aligned}$$

Bevis av d.l. for $S(\dot{\underline{\varepsilon}})$:

$$\text{V. siden: } \dot{\underline{R}}_p^n - \underline{\dot{\tilde{R}}}_p^n = \underline{\tilde{R}}_p^n (I + S(\underline{\varepsilon})) + \underline{\tilde{R}}_p^n S(\dot{\underline{\varepsilon}}) - \underline{\dot{\tilde{R}}}_p^n = \underline{\tilde{R}}_p^n S(\underline{\varepsilon}) + \underline{\tilde{R}}_p^n S(\dot{\underline{\varepsilon}}) = \underline{\tilde{R}}_p^n S(\underline{\omega}_{np}^p) S(\underline{\varepsilon}) + \underline{\tilde{R}}_p^n S(\dot{\underline{\varepsilon}})$$

$$\begin{aligned} \text{H. siden: } \underline{R}_p^n S(\underline{\omega}_{np}^p) - \underline{\tilde{R}}_p^n S(\underline{\omega}_{np}^p) &= \underline{\tilde{R}}_p^n (I + S(\underline{\varepsilon})) (S(\underline{\omega}_{np}^p) + S(\delta \underline{\omega})) - \underline{\tilde{R}}_p^n S(\underline{\omega}_{np}^p) \\ &= \underline{\tilde{R}}_p^n S(\underline{\varepsilon}) S(\underline{\omega}_{np}^p) + \underline{\tilde{R}}_p^n S(\delta \underline{\omega}) \end{aligned}$$

Satt sammen:

$$\underline{\tilde{R}}_p^n S(\underline{\omega}_{np}^p) S(\underline{\varepsilon}) + \underline{\tilde{R}}_p^n S(\dot{\underline{\varepsilon}}) = \underline{\tilde{R}}_p^n S(\underline{\varepsilon}) S(\underline{\omega}_{np}^p) + \underline{\tilde{R}}_p^n S(\delta \underline{\omega})$$

$$S(\dot{\underline{\varepsilon}}) = S(\underline{\varepsilon}) S(\underline{\omega}_{np}^p) - S(\underline{\omega}_{np}^p) S(\underline{\varepsilon}) + S(\delta \underline{\omega})$$

$$\begin{aligned} S_\delta &= \begin{bmatrix} 0 & -\delta_z & \delta_y \\ \delta_z & 0 & -\delta_x \\ -\delta_y & \delta_x & 0 \end{bmatrix}, \quad S_\omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad S_\varepsilon = \begin{bmatrix} 0 & -\varepsilon_z & \varepsilon_y \\ \varepsilon_z & 0 & -\varepsilon_x \\ -\varepsilon_y & \varepsilon_x & 0 \end{bmatrix} \\ S_\varepsilon S_\omega - S_\omega S_\varepsilon + S_\delta &= \begin{bmatrix} 0 & \varepsilon_y \omega_x - \varepsilon_x \omega_y - \delta_z & \varepsilon_z \omega_x - \varepsilon_x \omega_z + \delta_y \\ \varepsilon_x \omega_y - \varepsilon_y \omega_x + \delta_z & 0 & \varepsilon_z \omega_y - \varepsilon_y \omega_z - \delta_x \\ \varepsilon_x \omega_z - \varepsilon_z \omega_x - \delta_y & \varepsilon_y \omega_z - \varepsilon_z \omega_y + \delta_x & 0 \end{bmatrix} \end{aligned}$$

D.l. for $\underline{\varepsilon}$ skrevet ut:

$$\begin{aligned} \dot{\varepsilon}_x &= \varepsilon_y \tilde{\omega}_z - \varepsilon_z \tilde{\omega}_y + \delta \tilde{\omega}_x \\ \dot{\varepsilon}_y &= \varepsilon_z \tilde{\omega}_x - \varepsilon_x \tilde{\omega}_z + \delta \tilde{\omega}_y \\ \dot{\varepsilon}_z &= \varepsilon_x \tilde{\omega}_y - \varepsilon_y \tilde{\omega}_x + \delta \tilde{\omega}_z \end{aligned}$$

Dette kan også skrives

$$\dot{\underline{\varepsilon}} = -S(\underline{\omega}) \underline{\varepsilon} + \delta \underline{\omega}$$

Vedlegg B

Matlab-filer

- simulator.m (hovedprogram)
- funksjon.m
- Rpn.m
- skjev.m

simulator.m

```
% ----- Masteroppgave -----
% "Treghetsnavigassjon med smarttelefon"
%
%          Universitetet i Oslo
%      UNIK - Universitetssenteret på Kjeller
%          Hågen Kyllø
%          2012 06 11
% -----

% Simulering av sensordata fra banegenerator
% Simulering av navigasjonligninger

clear all; clc; close all;

% Tidsintervaller
t_0=1;           % start
t_1=200;
t_2=400;
t_3=600;
t_4=800;
t_5=1000;
t_6=1200;
t_7=1400;
t_8=1600;       % slutt

Ts=0.01;         % Time step
N=t_8;           % Total number of samples for planar movement
t_vec=(0:N)*Ts;  % Tidsvektor: fra 0-1600= 1601 celler

% Sette vektorer og initialverdier
f_n=zeros(3,1); % spesifikk kraft i n-ramma
w_p_n=zeros(3,1); % vinkelhastighet i n-ramma

% Setter matriser
Macc=zeros(3,N); % matrise - akselerarasjon
Mvel=zeros(3,N); % matrise - hastighet
Mpos=zeros(3,N); % matrise - posisjon
Momega=zeros(3,N); % matrise - vinkelhastighet
Mtheta=zeros(3,N); % matrise - vinkelposisjon

% ----- DETERMINISTISK LØSNING -----
f=(0.3/(2-(1/pi)*sin(pi*2)))*pi; % Spesifikk kraft 0,471238 m/s^2
w=pi; % w*2=2pi -> w=pi
a=zeros(1,t_1); % Initialverdi akselerasjon
v=zeros(1,t_1); % Initialverdi hastighet
p=zeros(1,t_1); % Initialverdi posisjon

for i=1:200
p(i)=(f/w)*((t_vec(i)-(1/w)*sin(w*t_vec(i)))-(t_vec(t_0)-
(1/w)*sin(w*t_vec(t_0))));
```

```

v(i)=-(f/w)*((cos(w*t_vec(i)))-(cos(w*t_vec(t_0))));
a(i)=f*((sin(w*t_vec(i)))-(sin(w*t_vec(t_0))));
end

for i=1:200    % t_0 -> t_1: beveges i x-retning fra 0 til 0,3m
    Macc(1,i)=a(i);
    Mvel(1,i)=v(i);
    Mpos(1,i)=p(i);
end

for i=t_1:t_2    % t_1 -> t_2 (2,00 - 4,00): pausering i 1. hjørne
    Mpos(:,i)=Mpos(:,t_1);
end

for i=1:200    % t_2 - t_3: beveges i y-retning fra 0 til 0,3m
    Macc(2,i+t_2)=a(i);
    Mvel(2,i+t_2)=v(i);
    Mpos(2,i+t_2)=p(i);
    Mpos(1,i+t_2)=Mpos(1,t_2);
end

for i=t_3:t_4    % t_3 -> t_4: pausering i 2. hjørne
    Mpos(:,i)=Mpos(:,t_3);
end

for i=1:200    % t_4 -> t_5: beveges i x-retning fra 0,3 til 0m
    Macc(1,i+t_4)=-a(i);
    Mvel(1,i+t_4)=-v(i);
    Mpos(1,i+t_4)=Mpos(1,t_4)-p(i);
    Mpos(2,i+t_4)=Mpos(2,t_4);
end

for i=t_5:t_6    % t_5 -> t_6: pausering i 3. hjørne
    Mpos(:,i)=Mpos(:,t_5);
end

for i=1:200    % t_6 -> t_7: beveges i y-retning fra 0,3 til 0m
    Macc(2,i+t_6)=-a(i);
    Mvel(2,i+t_6)=-v(i);
    Mpos(2,i+t_6)=Mpos(2,t_6)-p(i);
    Mpos(1,i+t_6)=Mpos(1,t_6);
end

for i=t_7:t_8    % 1400-1600, pausering i 4. hjørne
    Mpos(:,i)=Mpos(:,t_7);
end

g__n=[0 ; 0 ; -9.81];
for i=t_0:N
    Macc(:,i)=Macc(:,i) + g__n;    % gravitasjon legges på z-aksen
end

VRW=0.0033;          % Velocity Random Walk
ARW=0.0187;          % Angular Random Walk

```

```

Q_a=diag([VRW , VRW , VRW]);
Q_g=diag([ARW , ARW , ARW]);
b__a=0.001962;           % Bias til akselerometeret
b__g=0.01443;           % Bias til gyrometeret

ft__p=zeros(3,N); % spesifikk kraft - preallokering
wt_p_np=zeros(3,N); % vinkelhastighet - preallokering

for i=t_0:N
    ft__p(:,i)=Macc(:,i)+Q_a*randn(3,1)+b__a; % Trekker hvit støy
    wt_p_np(:,i)=Momega(:,i)+Q_g*randn(3,1)+b__g;
end

% ----- Plot banegenerator -----
% ----- simulert omløp -----
% figure()
% clf
% hold all, grid on, axis equal;
% title('Banegenerator: plot av simulert omløp')
% axis([-0.05 0.35 -0.05 0.35])
% xlabel('x-akse')
% ylabel('y-akse')
% plot((Mpos(1,t_0:t_1)), (Mpos(2,t_0:t_1)), 'k', 'LineWidth', 1.5);
% plot((Mpos(1,t_2:t_3)), (Mpos(2,t_2:t_3)), 'k', 'LineWidth', 1.5);
% plot((Mpos(1,t_4:t_5)), (Mpos(2,t_4:t_5)), 'k', 'LineWidth', 1.5);
% plot((Mpos(1,t_6:t_7)), (Mpos(2,t_6:t_7)), 'k', 'LineWidth', 1.5);

% ----- plot banegenerator -----
% ----- posisjon til x-akse i forhold til tid -----
% figure()
% clf
% hold all, grid on, title('Banegenerator plot x-akse: (- posisjon) (:
hastighet) (-- akselerasjon)')
% xlabel('tid')
% ylabel('x-akse')
% set(0, 'DefaultAxesColorOrder', [0 0 0], ...
%       'DefaultAxesLineStyleOrder', '--|:|-' )
% plot((t_vec(t_0:t_8)), (Macc(1,t_0:t_8)), 'LineWidth', 1.5);
% plot((t_vec(t_0:t_8)), (Mvel(1,t_0:t_8)), 'LineWidth', 1.5);
% plot((t_vec(t_0:t_8)), (Mpos(1,t_0:t_8)), 'LineWidth', 1.5);

% ----- plot banegenerator -----
% ----- vinkelhastighet i forhold til tid -----
figure()
clf
hold all, grid on,
title('Banegenerator: simulert vinkelhastighet')
axis([0 16 -0.3 0.3])
xlabel('tid')
ylabel('vinkelhastighet')

```

```

set(0,'DefaultAxesColorOrder',[0 0 0],...
    'DefaultAxesLineStyleOrder','-')
plot((t_vec(t_0:t_8)), (Momega(1,t_0:t_8)), 'k');

% ----- plot banegenerator -----
% ----- simulert spesifikk kraft med støy -----
figure()
clf
% hold all, grid on,
% title('Banegenerator: simulert akselerasjon med støy')
% axis([0 16 -0.5 0.5])
% set(0,'DefaultAxesColorOrder',[0 0 0],...
%     'DefaultAxesLineStyleOrder','-')
% xlabel('tid')
% ylabel('akselerasjon')
% plot((t_vec(t_0:t_8)), (ft__p(1,t_0:t_8)));

% ----- plot banegenerator -----
% ----- simulert vinkelhastighet med støy -----
figure()
clf
hold all, grid on,
title('Banegenerator: simulert vinkelhastighet med støy')
axis([0 16 -0.3 0.3])
set(0,'DefaultAxesColorOrder',[0 0 0],...
    'DefaultAxesLineStyleOrder','-')
xlabel('tid')
ylabel('vinkelhastighet')
plot((t_vec(t_0:t_8)), (wt_p_np(1,t_0:t_8)));

% ----- Navigasjonssimulering -----
x=[Mpos;Mvel;Mtheta]; %
xt=zeros(6,N); % x tilde - preallokering
u=[ft__p;wt_p_np]; % pådrag
R_p_n=zeros(3,3,N); % Rotasjonsmatrise

for k=t_0:N-1
    xt(:,k+1)=xt(:,k)+funksjon(xt(:,k),u(:,k),R_p_n(:, :, k))*Ts;
    R_p_n(:, :, k+1)=R_p_n(:, :, k)+Ts*Rpn(skjev(wt_p_np));
end

% ----- plot banegenerator -----
% ----- tilstand xt(1:3) -----
figure()
clf
% hold all, grid on;
% title('Navigasjonsligninger: xt(1:3), -|--|-.')
% axis([0 16 -15 10])
% set(0,'DefaultAxesColorOrder',[0 0 0],...
%     'DefaultAxesLineStyleOrder','-|--|-.')
% xlabel('x-akse')

```

```

% ylabel('y-akse')
% plot((t_vec(t_0:t_8)), (xt(1,t_0:t_8)), 'k');
% plot((t_vec(t_0:t_8)), (xt(2,t_0:t_8)), 'k');
% plot((t_vec(t_0:t_8)), (xt(3,t_0:t_8)), 'k');

% ----- plot banegenerator -----
% ----- tilstand xt(1:3) -----
% figure()
% clf
% hold all, grid on;
% title('Navigasjonsligninger: xt(4:6) -|--|-.')
% axis([0 16 -15 10])
% set(0, 'DefaultAxesColorOrder', [0 0 0], ...
%       'DefaultAxesLineStyleOrder', '-|--|-.')
% xlabel('x-akse')
% ylabel('y-akse')
% plot((t_vec(t_0:t_8)), (xt(4,t_0:t_8)), 'k');
% plot((t_vec(t_0:t_8)), (xt(5,t_0:t_8)), 'k');
% plot((t_vec(t_0:t_8)), (xt(6,t_0:t_8)), 'k');

```


funksjon.m

```
function [ f ] = funksjon(x,u,R)
% "funksjon" brukes i løkke for
% simulering av navigasjonsligninger

g=[0;0;-9.81];           % gravitasjon
f(1:3,1)=x(4:6);         %
f(4:6,1)=R*u(1:3)-g;     %

end
```

Rpn.m

```
function [ Rpn ] = Rpn( w )
% Brukes i simulering av navigasjonsligningene

Rpn=[1 sin(w(1))*tan(w(2)) cos(w(1))*tan(w(2));...
     0 cos(w(1)) -sin(w(1));...
     0 (sin(w(1)))/(cos(w(2))) (cos(w(1)))/(cos(w(2)));
     ];

end
```

skjev.m

```
function [ Mw ] = skjev( w )
% Tar en vektor med 3 parametre og setter dem inn
% i en skjevsymmetriskmatrise 3x3
Mw=[ 0      -w(3)    w(2);
     w(3)    0      -w(1);
     -w(2)   w(1)    0;
     ];

end
```